Manufacturing Process Planning Based On Machining Capability Profiles

Afsharizand, Behnood

Manufacturing Process Planning Based On Machining Capability Profiles

submitted by

Behnood Afshari Zand

for the degree of Doctor of Philosophy

of the

University of Bath

Department of Mechanical Engineering

July 2018

Declaration of authorship

I am the author of this thesis, and the work described therein was carried out by myself personally ....

Acknowledgements

Firstly, I would like to express my sincere gratitude to my PhD supervisors Dr. Aydin Nassehi, Prof. Stephen Newman and Dr. Vimal Dhokia for the continuous support of my PhD study and related research, for motivation, and immense knowledge. Their guidance helped me in all the time of research and writing of this thesis.

Besides my PhD supervisors, I would like to thank my PhD colleagues: Dr. Reza Imani, Dr. Joseph Michael Flynn, Dr. Wesley Essink and Dr. Alborz Shokrani for their insightful comments and encouragement and friendships.

My sincere thanks also goes to STEPMAN project committee, who funded this research and provided me an opportunity to join to Advanced Machining Processes and Systems (AMPS) research group in Bath.

Last but not the least, I would like to thank my family and friends for supporting me spiritually throughout writing this thesis and my life in general.

Abstract

Manufacturing resources degrade continuously throughout their life. This may be due to external factors such as wear and vibration, or operator induced factors such as the selection of incorrect cutting parameters. Planning manufacturing processes based on nominal machine data, can as a consequence result in the generation of inefficient and infeasible manufacturing instructions and out-of-tolerance parts. A method to accurately represent the actual capability of resources is needed to effectively model the manufacturing resources to enable more accurate process planning.

In this research a new framework for macro and micro process planning of CNC machining processes is proposed. The framework is based on a computer aided process planning system based on actual machine capability entitled CAPPable.

Machining errors affecting the overall health of machines have been reviewed and identified. STEP-NC has been used to model the machining resources and their associated errors. A manufacturing capability profile has been designed in which it is possible to store the values which reflect the degradation of machining resources.

CAPPable has been implemented as a STEP-Compliant prototype CAPP system for machining and validated on micro and macro levels. It has been demonstrated that using this framework, the current capability of resources can be accurately represented and can improve process planning effectiveness compared to using nominal manufacturing resource information. Through implementation of this framework, the capability of manufacturing resources can utilise resources to a far greater extent than currently possible. CAPPable has also been extended for use to generate improved part setup location routines for CNC machining.

Contents List

List of Figures

List of Tables

Nomenclature

ANN	Artificial Neural Network
AP	Application protocol
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CAPP	Computer Aided Process Planning
CAPPable	Process planning based on machining capability profiles
CD	Committee Draft
CNC	Computer Numerical Control
DOF	Degree of Freedom
EA	Evolutionary Algorithm
GA	Genetic Algorithm
IGES	The Initial Graphics Exchange Specification
IR	Integrated Resources
IS	International Standard
ISO	International Organisation for Standardisation
MCP	Machining Capability Profile
NIST	National Institute of Standards and Technology
PKM	Parallel Kinematics Machine
PSO	Particle Swarm Optimisation
STEP-NC	STEP-compliant Data Interface for Numeric
STEP	International Standard for Product Data Exchange
TS	Technical Specification
UML	Unified Modelling Language
WD	Working Draft
XML	eXtensible Markup Language

Definitions

Application programming interface: Application programming interface is a set of routines, protocols, and tools for building software applications.

CNC machining system: The organisation of various manufacturing resources responsible for executing manufacturing decisions.

Current state of manufacturing capability: A set of capability that a machine tool can deliver according to the latest condition of its resources.

Data model: A language to describe constructs for expressing intent.

Effective process plan: A process plan that can deliver the required geometric tolerances stated in the design of a part.

Entity: A class of information defined by common properties.

Entity data type: A representation of an entity. An entity data type establishes a domain of values defined by common attributes and constraints.

Framework: A framework is an abstraction in which software providing generic functionality can be selectively changed by additional user-written code.

G&M code: A language for describing CNC interpretable syntax for generating servo controlled axis movements.

Healthy machine tool: A machine tool that its performance has been checked though a series of tests against the machine tool specifications.

Interoperability: The ability to seamlessly transfer information from one CAD/-CAM/CNC system to another, while maintaining the integrity of the information.

Java class: A Java class is a template for creating different objects which defines its properties and behaviours. Java class objects exhibit the properties and behaviours defined by its class. A class can contain fields and methods to describe the behaviour of an object.

Java interface: The contact point between a class and the outside world. When a class `implements` an interface, it promises to provide the behaviour defined in that interface.

Kinematics: Is a branch of mechanics that describes the motion of points, bodies (objects), and systems of bodies (groups of objects) without considering the mass of each or the forces that caused the motion

Machine tool capability: A set of capability that a machine can provide at a particular point of time.

Macro process planning: Macro process planning is a process planning stage which concerned with identifying the main tasks and their best sequence and the type of manufacturing processes.

Manufacturing resources: Physical or virtual elements contributing execution of manufacturing activities such as mechanical resources and cutting tools.

Micro process planning: Micro process planning is a detailed process planning level in which the process planner determines the workpiece setup, machining strategies, cutting tools and cutting conditions.

Quasi-static error: This error varies slowly with time and depends primarily on the structure of the machine tool.

Robots singularity: is a point in the workspace where the robot loses its ability to move the end effector in some direction no matter how it moves its joints.

Semantics: The implied meaning of data within a specific context with respect to their role in a system.

Squareness: is the difference between the inclination of the reference straight line of the trajectory of the functional point of a linear moving component with respect to its corresponding principal axis of linear motion and (in relation to) the inclination of the reference straight line of the trajectory of the functional point of another linear moving component with respect to its corresponding principal axis of linear motion.

Straightness: is value of the largest positive straightness deviation added to the absolute value of the largest negative straightness deviation (with respect to any previously defined reference straight line).

Syntax: The rules that regulate the format of representation of information within a context. Syntax rules define whether or not an statement is valid for the grammar of the language.

Unified manufacturing resource model: A conceptual representation of the manufacturing resources that provides a logical framework for making desired manufacturing decisions.

Chapter 1

Introduction

1.1 Background

Manufacturing technologies have continuously improved over the past sixty years in terms of multiple criteria, such as machine architectures, control systems, computer aided production and process planning systems (Chen et al., 2015). The introduction of computer numerically controlled (CNC) machines in the 1970s, changed production from manually driven machines to automated systems with different capabilities (Vichare et al., 2009). These CNC machines, which are value adding in manufacturing operations are usually used on a consistent basis over a number of years but degrade in performance throughout their life cycle. When operating within a manufacturing environment, machine tools need to stay in a working condition according to machine tool manufacturer specification, to produce high-quality finished products. Machine tool faults account for yearly economic losses of tens of billions of US dollars (Lee, 1995). A process planner can only produce parts which meet design requirements if he or she has enough knowledge about available machining resources. The knowledge of machine capability is normally gained over decades by the process planner, who generate NC programmes on the same machining resources.

This knowledge gained from machining experience is permanently at risk of getting lost or forgotten by changing operators or procedures. As a result of this loss, the quality of machined parts can be effected. Capturing and storing the actual capability of machine tools helps the decision-making process of machining a part in the aforementioned situation. Usually the machine tool encompasses a huge system consisting of many assisting subsystems and mechanisms. For example, a 3-axis CNC milling machine includes machine tool axes, work table, tool handling, controller and coolant system. Capturing the knowledge of the capabilities across these machining components is increasingly complicated due to lack of methods to define these manufacturing resources.

At present, in the process of part manufacturing, there has been great interest in automation (Wang et al., 2016). Despite significant efforts to develop fully automated process planning systems, most developed process planning packages still rely on the knowledge of process planners for decision-making. One reason for this is because on the shop-floor there are various types of machine tools, and each has different capabilities. In order to automate CAD/CAM/CNC transformation, there has to be a flexible system to select machine tools based on their capabilities and adjust the process parameters in the generated machining instructions.

This research aims to specify and realise a novel framework for generating process plans based on the actual capability of machine tools at the time of generation. The proposed research will take place between the CAM and CNC stage in the process chain of part manufacturing. The theoretical method in this research contains modelling of manufacturing resources, machine tool errors and kinematics of machines.

The thesis is structured in seven chapters. This chapter introduces the contents of the research including research context and aims and objectives. This is followed by a review of the current methods for generating process plans in Chapter 2. The specification and design of a methodology for creating a manufacturing capability profile containing the machine tool health parameters is covered in Chapter 3. The development of a prototype is provided in Chapter 4. Experimental research conducted to validate the developed prototype is discussed in Chapter 5. Discussions on the results and findings of the research are presented in Chapter 6. Finally, the conclusions of the research along with potential future work are provided in Chapter 7. An outline of this thesis structure can be seen in Figure 1-1.

1.2 Research context

Existing machine tools in the shop-floor environment may not always be able to deliver the required tolerances, as, machining operations are affected by various external factors such as vibration, temperature and operator errors. A process planning engineer can only reliably select a machine which can deliver an accurate finished part if he/she has knowledge of the current state and capability of that particular machine. In most manufacturing enterprises this knowledge is tacit and not recorded. Effective use of resources is thus only achieved with considerable expenditure of time and gathering of human expertise.

A new process planning system using captured machine tool health data has been proposed in this research to transform this intensive activity to a well-defined formal

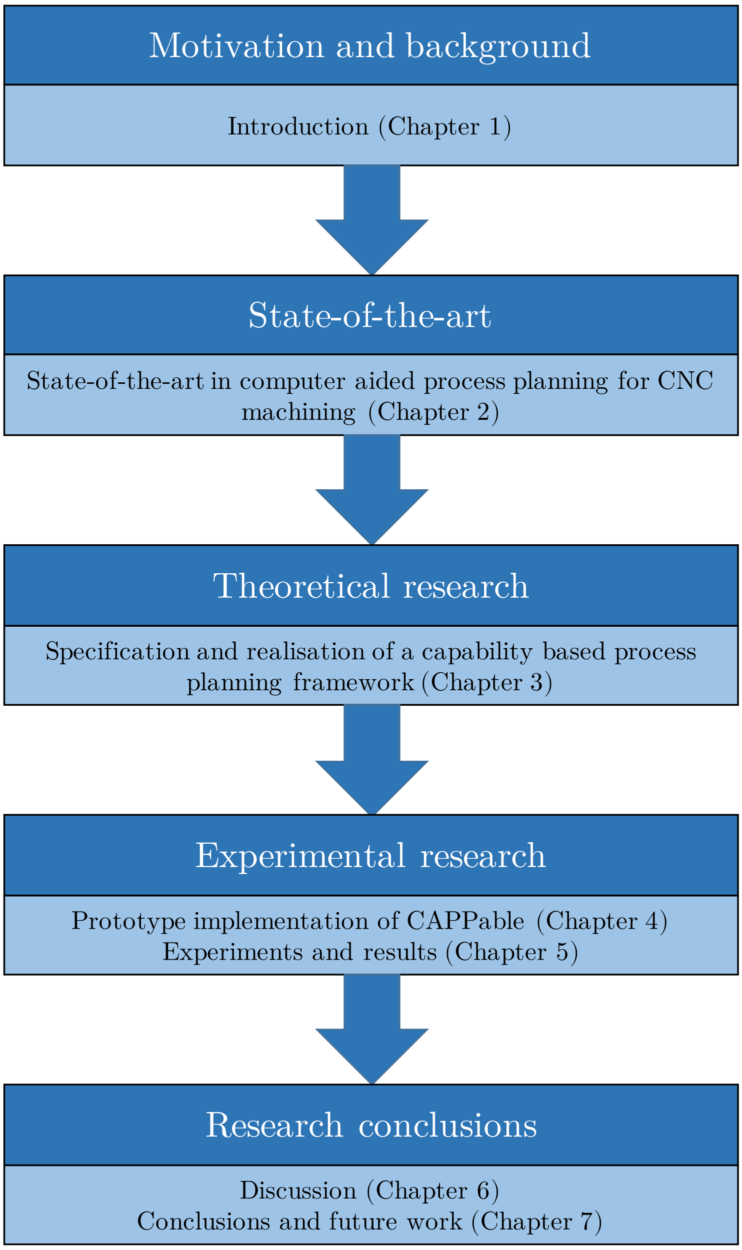


Figure 1-1: Structure of thesis chapters

process relying on explicit knowledge. The health data is structured as a capability profile that contains the entire body of information that a process planning system would need to decide whether it would be possible to produce a part on a specific machine and if so, the values that should be adjusted in the machine to achieve the desired results.

1.3 Aims and objectives

The aim of this research is **to specify and realise a framework to generate effective process plans based on the current state of the capabilities of manufacturing resources**. The effective process plans used in this context is a process plan that can deliver the required geometric tolerances stated in the design of a part.
This will be achieved through meeting the following objectives:

- Compilation of the state-of-the-art in process planning and management of machining capability knowledge through a comprehensive review of the literature to identify existing methods for capability based process planning and their gaps.

- Identification of the parameters that define machining capability (i.e. workspace, accuracy, position, fixturing and tooling). Quantification of the parameters that represent machine tool capability as explicit knowledge. Classification of the quantified parameters and their influence on a part process plan. Generation of a machine data model including a new data structure for machine tool capability and health.

- Development of a prototype.

- Validation of the proposed framework and the data model using the prototype.

This research is applicable to capability of a machine independent of machine used for mass production, batch production or flexible manufacturing. The ability to measure the capability of a machine over time will be highly valuable which ever the type of manufacturing is used.

1.4 Research boundaries and limitations

A number of research boundaries have been identified to allow the research to focus on the key issues of process planning based on the actual capability of a machine tool. These boundaries are illustrated in Figure 1-2, where the triangular shape represents the relevant research areas and the circle highlights the research boundaries. The areas within the circle are the major focus of this research, which are defined in the following sub-sections. The areas between the circle and the rectangle are considered in this research as the related areas. These areas have been reviewed in Chapter 2.

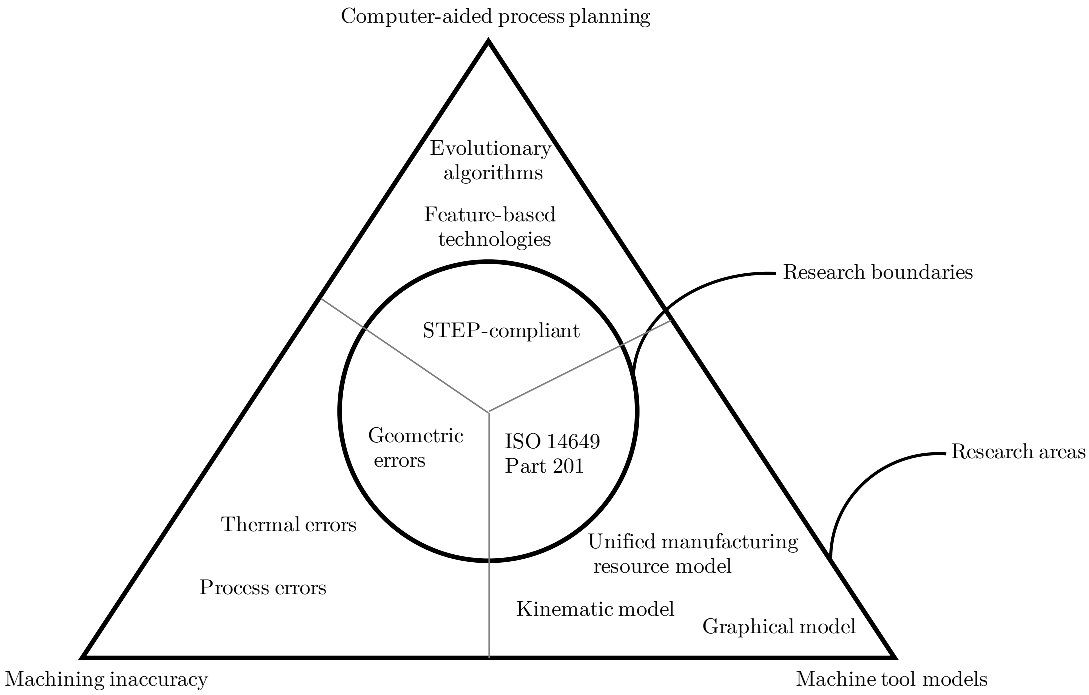


Figure 1-2: Research boundaries within the context of manufacturing

1.4.1 Computer-aided process planning

Various approaches are used to convert the raw materials into the finished products. Among these approaches, STEP-compliant has been selected as the most suitable approach for its great potential of modelling the complex information structure of process planning. Also, the extensibility of STEP can considerably reduce the chance of loosing information across CAD/CAM/CNC chain. The data structure of STEP also supports decision-making of manufacturability of parts. The structure of STEP has been covered comprehensively in Section 2.4. The other approaches such as feature-based approach and evolutionary algorithm has been reviewed in Sections 2.3.2 and 2.3.1.

1.4.2 Machining inaccuracy

Improving machine accuracy and selecting suitable machining processes are key to achieving high quality manufactured products. The importance of improving machine tool accuracy is well recognised due to the increasing demand for high precision parts (Ni, 1997). In addition, machine tool builders have been continuously searching for ways to improve the quality of their products and to reduce costs of the machines that they produce in order to stay competitive.

Generally, there are various factors which can affect machining accuracy such as temperature and vibration. The three major types of errors are: 1. geometric errors; 2. thermal errors; and 3. cutting-force induced errors. Factors which can affect machining accuracy are covered in Section 2.5.2. Geometric errors make up the major part of the inaccuracy of a CNC machine (Ramesh et al., 2000a) and can be assumed to remain constant through the machining of a single component. Thermal errors and the errors caused by cutting forces are more dynamic and variable than geometric errors.

Without loss of generalisation, this research focuses on geometric errors due to their importance in inaccuracy of machines (Wan et al., 2016) and the relative simplicity of estimating their effect on the machining process. Estimating the effects of the other two types of error would require dynamic and thermal modelling of machine tools which would add to the complexity without adding to the core contribution of this work.

1.4.3 Machine tool models

Machine tool models are conceptual representations of the real machine tools. There are a number of approaches to represent machine tools such as graphical simulation, kinematic representations and dynamic models. Vichare et al. (2009), proposed a unified manufacturing resource model to integrate the approaches that define the functional capability of a machine tool to produce a data model that can inform the process planning activity (Vichare et al., 2009). The other approach for representing the machine tool capability has been presented in ISO 14649-201 (2011). The model defined using the EXPRESS data modelling language used throughout the STEP series of standards (ISO 10303-1, 1994) to represent manufacturing resource capabilities. Graphical models provide visual confirmation of machineability based on the kinematic reachability of the machine. Although graphical models are extensively used in the commercial CAM software they provide limited information to process planners to make decisions based on the current machine tool health. Graphical models do not contain machine tool component capability such as, axes repeatability and axes positional accuracy. Their application is limited to simulation and verification of machine movements such as, tool path verification and tool accessibility checks. Section 2.5.1 provides a detailed review of these machine tool models.

1.5 Scope of research: areas of investigation

To achieve the research objectives outlined in Section 1.3, the research scope has been identified as follows:

(i). Review of the CAPP systems developed for machining capability: A comprehensive state-of-the-art review will be carried out, identifying various capability based process planning systems currently developed together with their approaches. Techniques used to capture the actual capability of machining resources will be covered in this review. Among these techniques, kinematic models of machine tools will be the focus as it is identified to be the most widely used method to assess machineability. This review will highlight a number of factors which will be utilised for developing the data model of undertaken research method.

(ii). Development of a process planning framework to capture the real capability of machines: Process planning based on the nominal capability of manufacturing resources can result in out-of-tolerance parts. Process planning output can be improved by considering the design specification as well as the actual capability of the available machining resources. A proposed process planning framework (CAPPable) will be introduced to decide whether it is possible to machine a part on a specific machine. The machining capability profile (MCP) will be introduced as a part of this development. Each MCP file contains capability data points which can be incorporated with CAPPable. This system works solely based on the machine tool models and does not investigate the machineability of materials.

(iii). Application of the proposed framework on three-axis machines: To present this body of work, a series of simulation studies will be conducted to demonstrate the proposed process planning framework at macro level. The process planning approach consisting of machine tool models will be evaluated by a series of tests. The feasibility of machining a sample part on a specific machine tool will be investigated through the CAPPable framework. The result of these tests will be generated automatically from CAPPable.

(iv). Application of the proposed framework on a PKM: Capturing the real capability of a parallel kinematic machine will be targeted on simulation software. This case study will demonstrate the proposed framework in detail at the micro process planning level. The best part location will be optimised considering the translational error along PKM legs. The fitness of the generated locations/particles has been evaluated with particle swarm.

(v). Validation of the proposed framework using the ball bar test: A series of ball bar tests have been done on a serial kinematic machine to validate the proposed process planing framework. Captured squareness and straightness errors have been modelled mathematically. A kinematic model of a CNC machine has been used to find the best part location on machine tool bed.

1.6 Research methodology

The research problem this work addresses is defined as **the lack of explicit knowledge of the current capability of machines during process planning**. To test the validity of this problem definition and the proposed solution, the hypothesis has been defined as: "*A process planning system that takes the current status of resources into account will produce results that are as good as or better than process planning based on nominal resource information*". This research is thus based on testing a hypothesis through the realisation and validation of a framework showing that solving the identified problem using the proposed technique, will produce better process plans.
Consequently, the deductive research method has been selected to test the hypothesis as follows:

1. Various standards and literatures are reviewed to specify the research gaps in manufacturing process planning and identify the capability parameters which can affect process planning outputs.

2. Based on the gaps identified in the literature a framework for a computer aided process planning system based on the actual available machine capability is defined and the functional requirements of CAPPable are identified.

3. A data modelling language is used to enable explicit expression of the actual machine capability information. The produced data model is incorporated within the CAPPable framework. Machine tool degradation data is then used to create knowledge instances representing physical machine conditions for CNC equipment.

4. A prototype implementation of CAPPable framework will receive product specifications and perform capability checks on the available manufacturing resources.

5. Comparative tests on selected machine tools will use CAPPable prototype to validate the proposed system in macro process planning. The proposed system is tested on the serial kinematic machines to check the feasibility of the process plans.

6. Error modelling of squareness and straightness using the ball bar test data to validate the CAPPable prototype on a three-axis serial machine. The best part location has been determined based on the CNC health profile.

Chapter 2

State-of-the-art in capability based computer aided process planning for CNC machining

2.1 Introduction

Computer-aided process planning (CAPP) has been the subject of many research projects conducted both by industrialists and academics over the past forty years, resulting in an enormous body of publications. This chapter reviews the various approaches undertaken by researchers, and provides a brief assessment of these approaches to highlight the research gaps on consideration of actual resource capability.

2.2 Current approaches for CAPP

In part machining, process planning is the act of preparing detailed sequences of machining operation instructions to transform an engineering design to a final product (Xu et al., 2011). Computer-aided process planning (CAPP) is the use of computer technology to aid in process planning of a part or product in manufacturing (Alting and Zhang, 1989). The input to the process planning system is the design of a part and in case of CAPP, this is supplied in the digital format of a Computer Aided Design (CAD) file. The output is the set of information that is required to manufacture the product and make sure that it meets the original design specifications.

A complete machining process plan should thus contain machining parameters, setups and tools required for production as well as the machining instructions required to make the part. Figure 2-1 shows the general approach to CAPP (Qiao et al., 1994). In manual process planning, a human process planner receives part description information such as dimension, tolerance and surface quality (Chang and Wysk, 1984). Based on this information and their knowledge of available machining resources, they must recognise and extract manufacturing features from the engineering drawing. Next, based on geometric and tolerancing considerations, they select suitable machining operations.

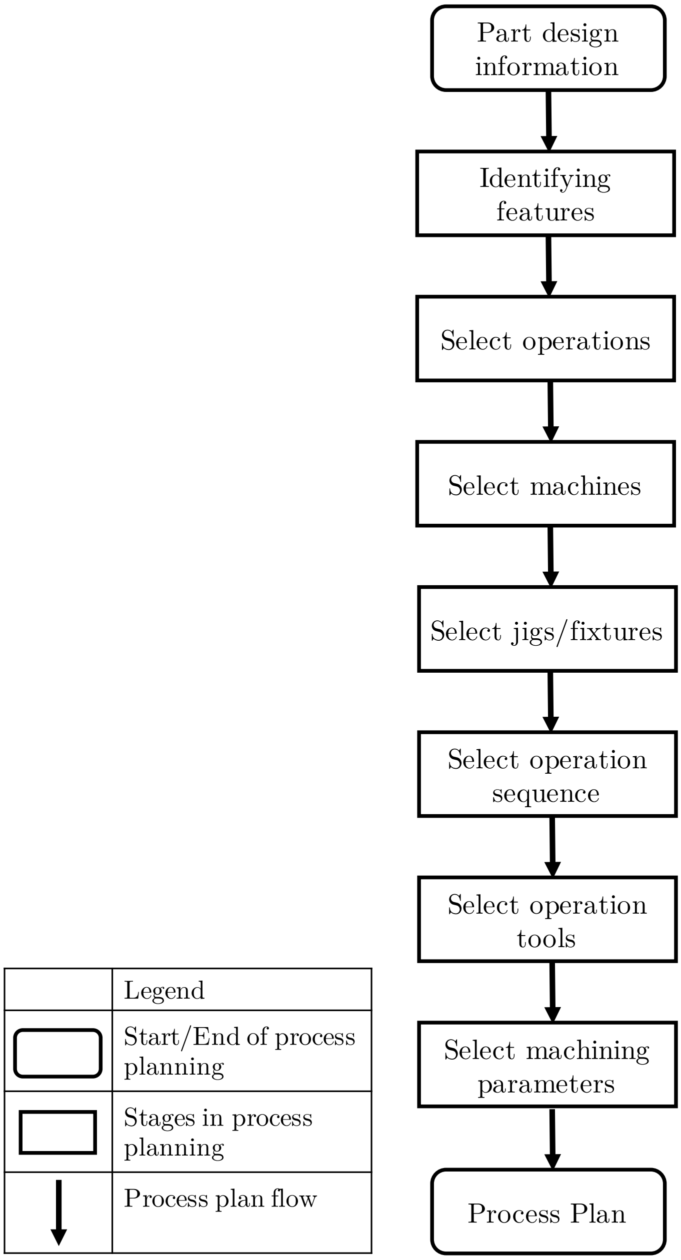


Figure 2-1: Process planning flow and stages (Qiao et al., 1994)

Manual process planning is time consuming and expensive in terms of technical expertise. This has been the major driver of CAPP research to capture the process planning knowledge in a computer based system that can be operated by people with less experience. Two general approaches have been proposed for CAPP: the variant approach; and, the generative approach (Alting and Zhang, 1989).

The variant approach to process planning is comparable with the traditional manual approach where a process plan for a new part is created by recalling, identifying and retrieving an existing plan for a similar part (known as a master plan), and making necessary changes for the new part (Alting and Zhang, 1989). In some variant systems parts are grouped into a number of part families, characterised by similarities in manufacturing methods and thus related to group technology. For each part family, a standard process plan, which includes all possible operations for the family, is stored in the system. An example of the variant approach is MIPLAN (Houtzeel, 1980). MIPLAN is a data retrieval system which retrieve process plans based on part code. By inputting a part code, parts with a similar code are retrieved. The process plan for each part is then displayed and edited by the user.

The variant approach is reliant on the availability of an appropriate information infrastructure that would allow the process plans for previous components to be stored (Yusof and Latif, 2014). The infrastructure should also allow the information to be quickly accessed and modified to match the requirements of the product for which a new process plan is required.

With this infrastructure in place, which usually takes the form of a database, the complicated activities and decisions take less time and labour to be processed. This is because the new part information is linked to already existing information. Furthermore, procedures can be standardised by adapting planners' manufacturing knowledge and structuring it to a company's specific needs and introducing them as validation rules in the information infrastructure (Ham et al., 2016).

The volume of the information and the sophisticated structure required to manage it, makes maintaining consistency in editing practices, and accommodation of various combinations of geometry, size, precision, material, quality, and machine tool loading. The biggest disadvantage of the variant approach, however, is that the quality of process plan still depends on the knowledge of a process planner (Yusof and Latif, 2014). Overall the advantages of the variant approach has been summarised by Tan and Khoshnevis (2000) as:

(i). The investment cost is lower and the development time is shorter. Especially for medium sized companies, who want to adapt their own process planning techniques (Tan and Khoshnevis, 2000).

(ii). The development costs and hardware costs are lower. Especially for small companies where the products do not vary significantly and process plans generate by human (Chryssolouris, 2006).

In the generative approach process plans are generated by means of decision logic, algorithm, and geometry based data to make various decisions for converting a part from raw material to its finished state. The rules of manufacturing and the equipment capabilities are stored in a database (Chryssolouris, 2006).

When using the system, a specific process plan for a part can be generated without human intervention. Process planning parameters can be imported in the format of text or graphic where the part data is collected from a CAD model. With the rapid development of computer intelligence, both academics and industrialists have gained interest in using graphical input for generative systems (Boer et al., 1990).

Technologies such as feature recognition, feature extraction and geometric reasoning have been adapted as an effort to interface CAPP with CAD. Many generative process planning systems have already been developed over the last 30 years such as EXCAP (Darbyshire and Davies, 1984) and AutoPlan (Patil and Pande, 2002). EXCAP was designed for turning operations to overcome the issue of reliance on human process planner. AutoPlan adapted the feature based design to map the identified features to the corresponding machining process. The biggest advantage of this approach is that the process plan is consistent and fully automated (ElMaraghy and Nassehi, 2014). SCSTO (Yusof and Case, 2010) was developed to generate machining instructions based on machining features to support the interactive generation of process plans utilising feature extraction. SCSTO was constructed using a structured methodology for its planning and object-oriented methods for its implementation. A prototype implementation of SCSTO showed that the new approach can eliminate the need for post processors.

The generative approach of process planning has became the emerging trend in CAPP studies towards automated and intelligent process planning systems (Xu et al., 2011).

To make decisions in generating the plans various technologies such as decision rules, optimisation algorithms (ElMaraghy, 2007), intelligent agents (Allen et al., 2005; Nassehi et al., 2006; Shen et al., 2006) and evolutionary algorithms (Li et al., 2010a) have been adopted by researchers. Effective generative process planning requires a complex decision-making process and as a result efficient algorithms and technologies are required to create such systems fit for function in a practical environment.

2.3 The evolution of CAPP

The idea of CAPP was firstly presented in 1965 (Niebel, 1965) where computers were used to assist humans to achieve better process plans in less time. One of the major pioneers of CAPP was Wysk (1977) who has submitted his PhD thesis in automated process planning and selection program: APPAS. The first CAPP system was developed in 1976 under the sponsorship of computer aided manufacturing international (CAM-I) (Cay and Chassapis, 1997). Since then, numbers of CAPP systems have been introduced, but none of them give practical solutions to manufacturers. The lack of knowledge about processes and resources made it difficult for CAPP developers to come up with a practical solution. Depending on the intelligence and complexity of the designed CAPP systems, these systems were divided into several groups. Alting and Zhang (1989) grouped and reviewed fourteen well-known CAPP systems designed by 1989, with general aspects of these systems such as approaches of implementation, methodologies and architectures discussed. ElMaraghy (1993) identified important key research issues in CAPP such as development of product definition, realising techniques for transforming design features and introduction technological data and knowledge base. Xu et al. (2011) reviewed the development of modern CAPP systems and categorised them based on the technologies adapted. Xu et al. (2011) defined 10 categories for CAPP systems. These systems were categorised as feature-based technologies, knowledge-based systems, artificial neural networks, genetic algorithm based systems, fuzzy set theory, petri nets, agent-based technology, internet-based technology, STEP-compliant and other emerging technologies.

The author has reviewed literatures specifically related to the categories of feature based CAPP systems, evolutionary algorithms in CAPP and STEP-compliant CAPP as these areas relate specifically to their research.

2.3.1 Feature-based technologies

The heart of any generative CAPP system is the engine that extracts machining features. This is because almost all CAPP systems function on the basis of features, or require features to be the input data such as SCSTO for turning (Yusof et al., 2009) and SFPS for milling (Suh et al., 2003a). Most of feature recognition systems developed for CAPP are based on three-dimensional solid models (Babic et al., 2008). Only few researches have been done on the reconstruction of objects from two-dimensional drawings (Wesley and Markowsky, 1981).

The main goal of feature recognition is to extract features at the appropriate level of abstraction from a given part model. This feature extraction needs to be sufficiently accurate and flexible to ensure that process planning system will yield useful results (Jha and Gurumoorthy, 2000). Feature-based CAPP systems enable process planners to translate traditional CAD files without being restricted to a limited set of predefined features. There are many different recognition methods developed for feature recognition, but the main drawbacks of them are high computational time, multiple interpretations, lack of machineability and high complexities with interacting features (Babic et al., 2008). Table 2.1 summarises the feature recognition methods used in CAPP systems.

Table 2.1: Feature recognition approaches used in CAPP

Methods	Criticisms in literature
Graph matching	It requires high computation time to deal with interacting features. Also, there is no guarantee that the recognised features can be machined using a 3-axis CNC (Zhang et al., 2014).
Volume decomposition	Since volume decomposition designed specifically for 3D volumes, this method has better efficiency in handling interacting features, but still suffers from expensive computations (Mascle et al., 2007).
Rule-based	It uses an artificial intelligence engine to run feature recognition. It is not a popular approach since it is impossible to define reasoning rules for all features and it does not cover all range of feature adjustments (Lam and Wong, 2000).
Neural network-based	It contains nodes and connections which use from learning ability to run feature recognition. Neural networks are time consuming because it requires a comprehensive feature definition language to yield useful results. (Lam and Wong, 2000).
Hybrid	Combining several advantages of the above methods attracted researchers to use from hybrid methods. For instance, Vandenbrande and Requicha (1993) combined rule-based and volume decomposition method adapted to cope with interacting features. Graph matching and volume decomposition approach has been applied by Rameshbabu and Shunmugam (2009) to recognise manufacturing features from 3D model data in STEP AP-203.

2.3.2 Evolutionary algorithms in process planning

Available CAPP systems significantly rely on human process planners to understand the rules involved in machining operations. When small changes are introduced into the manufacturing environment such as defining new tools or material, process plans need to be adjusted by manufacturing engineers. This is mainly because new materials can not be processed with the similar machining configurations such as feeds, speeds and cutting force. The most attractive feature of evolutionary algorithm (EA) is the flexibility of handling various kinds of objective functions with considering dynamic manufacturing changes. The main procedure for implementing EA-based approach are generating the initial population, evaluating the fitness of each individual and repeating the same steps until termination. In this section several EA-based tools used previously in CAPP have been reviewed.

Artificial neural network (ANN) is an established technique in artificial intelligence (Wang, 2013). One of the main advantage of an ANN is its ability to learn complicated relationships between machining parameters. ANN-based systems are mainly structured by these parameters: the neural net, training of learning rules, input code characteristic and output node characteristics (Prabhakar and Henderson, 1992). An ANN has several advantages over the other methods (Yue et al., 2002):

- It can ignore minor errors from input during learning or problem solving.
- It runs faster because the process does not involve searching to parse information
- ANN-based system improves by deriving rules or knowledge through training with examples

Knapp and Wang (1989) applied machine learning techniques based on if-then rules for process planning in an ANN-based system. The main limitations of this approach are complex computations, huge numbers of rules and manual outputs. Yue et al. (2002) concluded that ANN techniques for feature recognition used in process planning can eliminate these drawbacks, this is achieved by its ability to recognise incomplete features, slow computational speed, and its learning capability.

However, there are certain limitations with the use of ANN for both feature recognition and CAPP, namely, a limited range of features can be recognised, together with a limited range of component types, and also a lack of robustness (Yue et al., 2002).

Automatic acquisition of process planning knowledge has been implemented previously (Knapp and Wang, 1992). Parts to be machined are decomposed into machining features such as slots, holes and planes. Each feature type is associated with a set of characterising attributes such as dimensions and tolerances. Every feature is represented by a vector that its elements identify the feature type and its attribute values. Neural networks have been used to find the best sequence of operations for machining each feature of the part independently.

Santochi and Dini (1996) have demonstrated the applicability of neural networks for the automated selection of technological parameters of tools in turning such as angles of cutting edge and corner radius. Neural networks have been used as a programming tool for the selection of the mentioned parameters.

The classification and coding of parts for group technology applications continue to be labour intensive and time-consuming processes. A neural network system (Kaparthi and Suresh, 1991) has been proposed for classifying parts based on bitmaps of the part drawings. A neural network system is used to generate part geometry-related digits of the Opitz code from bitmaps of part drawings. Opitz is a coding system used to form groups in group technology. It has been stated that the proposed system is a useful tool for the automatic generation of shape-based classes and codes.

Neural networks have been used extensively for selecting optimal cutting conditions (Park et al., 2000), and can either be trained under supervision (i.e. by providing feedback on the learning progress) or be unsupervised. Park et al. (2000) used incremental supervised learning (Rao et al., 2014), which enables nodes to improve cutting conditions. When a new cutting condition is introduced to the system which yields better results, the proposed algorithm can adapt itself to this change.

Genetic Algorithms (GAs) are meta heuristic problem solving techniques that are used in many types of industries and have many applications in solving combinatorial problems (Renner and Ekárt, 2003). Mathematical problems that arise in manufacturing processes are often nonlinear and multi-objective. These types of functions are computationally expensive to evaluate based on deterministic optimisation methods. Genetic algorithms (Holland, 1992) are stochastic in nature: a random initial population of results is generated and then an evolution process is used to generate subsequent generations. In each generation the fittest members of the population are selected and through the use of a cross over algorithm, combine their genes to create offspring. Random mutations reduce the probability of the population being stuck around local optima. The main advantage that GA-based approach has over other CAPP approaches is in the task of considering the multiple decision-making activities, i.e. machine tools, cutting tools, tool access directions, and the sequence among the operations. Therefore, the generated process plan is the best suited among the entire solution space, and the optimal process plan for a part can be selected. The main drawback of the GA-based approach is that it could increase the complexity when manufacturing resources are not identical and work with different capabilities (Li et al., 2005). Also, the amount of memory or computation time required may become uneconomical when the problem grows beyond a certain size.

GA has been applied in CAPP by many researchers. GA-based process planning has been utilised to minimise production costs in turning (Reddy et al., 1998) and milling (Shunmugam et al., 2000). Tool life, cutting force and cutting power have been modelled mathematically and then optimised through the use of GAs. GA has been used in selecting and sequencing operations based on the machining features defined by Zhang (1997), and mapping these features to the appropriate series of operations (Bierwirth and Mattfeld, 1999). The goal of such systems are, generally, minimising production cost, machine changes, setup changes and tool changes.

Among other meta heuristic methods, particle swarm optimisation (PSO) has demonstrated promising results with increasing the chance of finding an optimal process plan in a short amount of time (Guo et al., 2009). Like GA, PSO relies on populations but each individual in a PSO is given a velocity index to move in the direction of the fittest members of the population. Particles are given an inertia to make sure that they do not converge too quickly at a local optimum and explore the search space. The distributed nature of PSO (and the ability to make the necessary calculations in a decentralised manner), generally reduces the amount of memory and computation time required compared to other evolutionary algorithms such as GA and ANN (Shi and Eberhart, 1999). PSO has been used in process planning successfully by researchers in areas such as, setup planning (Kafashi et al., 2012), tool wear monitoring (Gupta et al., 2016) and operation sequencing (Li et al., 2013).

Most case studies on EA-based process planning are concerned with the optimal selection of machining parameters such as cutting speed, feed and depth of cut. This is most likely because such problems are relatively easy to model as mathematical optimisation problems with clear definitions of objective functions and constraints. In addition, a number of CAPP studies have been concerned with improving the solution quality and reducing the computation time (Su et al., 2015). These studies assume that machining economics involve the optimal selection of machining parameters subjected to particular resource constraints such as maximum machine feed and speed. But, the optimal process plan can only be obtained by considering the manufacturing resource availability and accuracy. The range of cutting feed and speed is normally defined by machine tool manufacturers. The feasibility of cutting within this range is still based on the knowledge of process planner.

2.4 STEP-compliant CAPP

As mentioned earlier, process planning is a knowledge intensive task and structuring the relevant data for supporting process planning decisions is a major challenge. Furthermore, in design and manufacturing, data incompatibility is very costly. The National Institute of standards and Technology (NIST) estimated these costs at $90 billion in the USA (Brunnermeier and Martin, 1999). The reason for this cost is that each system requires the information to be structured in a different format, so the same items of information have to be manipulated multiple times. A major survey on the collaboration and interoperability market conducted by Longview Advisor shows that 100% of original equipment manufacturers (OEMs) are exchanging design/manufacturing data with suppliers (LongView Advisors, 2010). This takes a huge effort, and increases the risk of errors. Many authorities including international standard organisations have been seeking a solution to the problem of exchanging product data between dissimilar CAD/-CAM for decades. Such solutions could provide a suitable foundation for provision of information to CAPP systems.

Defining a neutral exchange file has been accepted to be a practical solution. As a result, several standards have been introduced such as SET (1989) in France, DIN 66301 (1987) in Germany and IGES (Smith et al., 1983) in the USA. But, these standards were only focused on geometric data exchange. Finally, STEP (ISO 10303-1, 1994) was introduced by International standard Organisation (ISO) as one International standard for all aspects of technical product data. The differences between two most common neutral CAD formats which are IGES and STEP are as follow:

An IGES file contains basic CAD information, namely, (i) Design features in 2D and 3D such as curves, surfaces, and wireframes, (ii) Drafting elements like lines and annotations, (iii) Finite element modelling elements, and (iv) Language and product definition data.

At first, STEP files covered the same product definition information as IGES, with the following additions of topology, tolerances, material properties, together with, other complex product data. The main objective of STEP is to provide a mechanism that is capable of describing product data, independent from any particular system. The application of STEP data exchange has been illustrated in Figure 2-2 in black arrows. In addition to STEP and IGES, other CAD neutral formats have been developed for exchaning design data such as JT (ISO 14306, 2012) and 3DXML (Versprille, 2005).

One of the key strengths of STEP is that it is extensible to represent every major CAD/CAM system. Recently, STEP-NC (ISO 14649-1, 2003) extension has been introduced to handle NC processes. Dashed arrows in Figure 2-2 shows the application of STEP-NC data exchange.

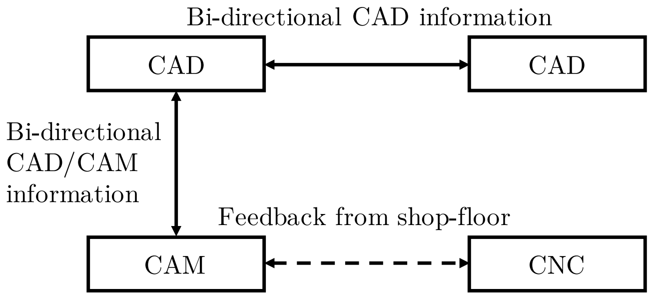


Figure 2-2: Application of STEP and STEP-NC across CAD/CAM/CNC (Xu, 2006)

2.4.1 STEP architecture and design

STEP aims to fully cover a product's data requirements throughout its life-cycle. The model is thus complicated and has been structured into various parts in five categories:

- Description methods: the EXPRESS language formalised in ISO 10303-11 (1994) is the modelling language used in STEP. EXPRESS defines an object-oriented like language for describing the data structure: concepts such as inheritance (entities inheriting their properties from another 'parent' entity) and various types of attributes are defined. The notion of methods (or actions) however is quite limited in EXPRESS compared to a modern Object Oriented language like C++ or Java. Realising this potential shortcoming for direct implementation of EXPRESS, STEP developers have specified bindings for C, C++ and Java to allow STEP compliant software development using these languages. Data modelling with EXPRESS has been well explained by Schenck and Wilson (1994), and it will be used later to introduce the proposed framework in this research in Chapter 3.

- Implementation methods: specify a mechanism that allows product data represented using the EXPRESS language, to support bidirectional information flow. For example, STEP Part 21 (ISO 10303-21, 2002) defines a format for recording and storing instances of entities modelled in EXPRESS using text files. Implementation methods define application data in EXPRESS that software developers can use to store and retrieve data pertaining to a population.

- Application protocols (APs): are a set of standards in ISO 10303 which specifies an application interpreted model providing data exchange for a particular family of products. APs are the parts of STEP intended to be applied in industry. For instance, a data exchange for casting processes has been defined in ISO 10303 part 223 (ISO 10303-223, 2008).

- Integrated resources (IRs): provide concepts to different application areas to use in their AP. IRs are normally divided into two groups: generic resources and application resource. Generic resources (ISO 10303-1, 1994) are fundamental definitions of a product such as geometry, materials and form features. Application resources (ISO 10303-1, 1994) are narrowly specialised for a group of product family such as finite element analysis and kinematics. Application resources can be adapted from the concepts defined in generic resources.

- Conformance testing: is divided into two series of STEP parts: conformance testing methodologies and test suites. Conformance testing methodology and framework (ISO 10303-1, 1994) describes testing procedure for implementation of various STEP parts. Test suite (ISO 10303-1, 1994) offers the set of abstract test cases necessary for conformance testing of an implementation of a AP. Figure 2-3 presents the structure of STEP containing all STEP protocols.

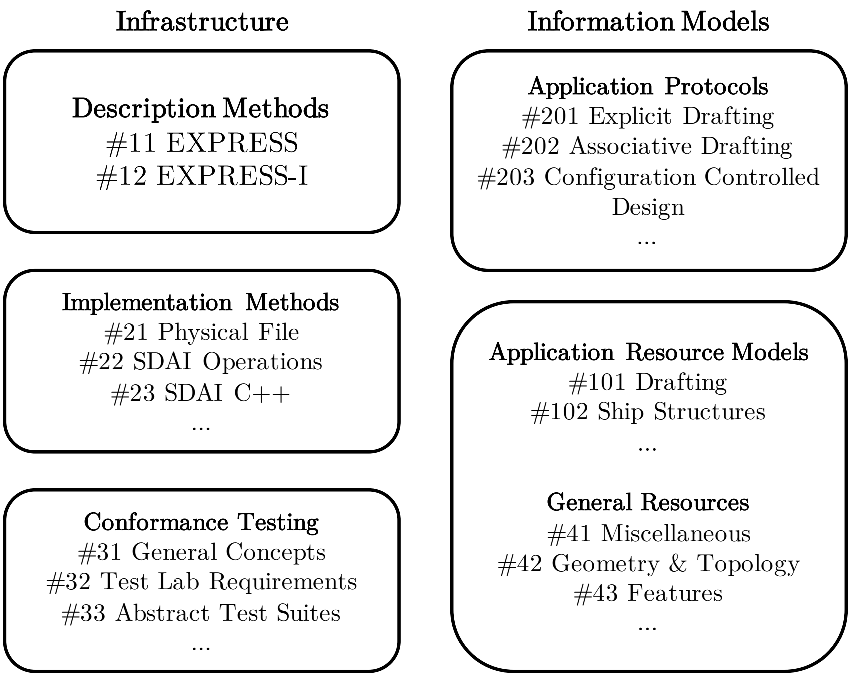


Figure 2-3: High level structure of STEP (Loffredo, 1999)

Using the structure mentioned above, it is possible to define data models for products and extend them to application protocols using integrated resources. The result will be a neutral data format which can be used by multiple systems.

2.4.2 STEP-NC structure

Thirty years of effort to develop STEP standards has led to the possibility of using standard data throughout the entire process chain of manufacturing enterprise. Low level G and M codes (ISO 6983-1, 1982) offered very limited information and semantic content. The main problem with G-codes is that they only provide low level machine axes movement and switching instructions in a predefined sequence, with no information given about the desired outcome of the machining. This means a part can be machined, however there is no way of checking if it adheres to the design, without referring back to the original drawing.

STEP-NC was thus proposed to alleviate this problem. The standard was firstly introduced in 2003 (ISO 14649-1, 2003) to replace the G&M codes standardised in RS274D and ISO 6983 (ISO 6983-1, 1982). STEP-NC provides a data model for unambiguously describing manufacturing tasks for all operations which are necessary to produce a finished part from raw material. The machining instructions are normally described by a sequence of these manufacturing tasks so called "workingsteps". This rich data model also allows bi-directional information transfer provides process planners with any modifications at the shop-floor which enables a better exchange of experience. The main advantage of STEP-NC is that process planners can use the same STEP-NC file for multiple machine tools. So, process planning does no longer cause bottlenecks for the manufacturing industry.

The ISO 14649 (ISO 14649-1, 2003) data models describe the machining requirements for CNC and how to present these requirements combined with product data defined by ISO 10303 (ISO 10303-1, 1994). Consequently, it can provide an excellent infrastructure for storing the process plan in a CAPP system. ISO 14649 part 10 (ISO 14649-10, 2004) specifies the process data that is common for all NC machining processes. These data models describe the interface between a CNC and the CAM software. Both geometric and technological information are included in a STEP-NC file. A healthy STEP-NC file uses ISO 14649 part 10 (ISO 14649-10, 2004) for part description combined with other STEP-NC parts. Process planner needs to identify the link between ISO 14649 part 10 (ISO 14649-10, 2004), and how it can be used together with various technology-specific parts in STEP-NC (Zhao et al., 2008). The description of these technology-specific parts has been listed in Table 2.2.

In a STEP-NC file, the HEADER contains general information about the program such as file name, author, date and organisation. The DATA section contains all the information about manufacturing tasks and geometries. STEP-based machining is normally start with a CAD model defined in STEP, followed by machining instructions provided by STEP-NC. The STEP-NC file known as part-21 file format is currently the most popular implementation method for EXPRESS defined STEP data. Figure 2-4 shows the structure of a part-21 file.

Table 2.2: Currently published parts of ISO 14649

ISO 14649 parts	Description	Current stage
part 1	overview and fundamental principles	IS
part 10	general process data	IS
part 11	process data for milling	IS
part 111	tools for milling machines	IS
part 12	process data for turning	IS
part 121	tools for turning machines	IS
part 13	process data for sink electrical discharge machining	IS
part 14	process data for wire electrical discharge machining	IS
part 15	contour cutting of wood and glass	CD
part 151	tools for contour cutting of wood and glass	CD
part 17	process data for additive manufacturing	WD
part 201	machine tool data for cutting processes	TS

Key: IS= International Standard, WD= Working Draft, CD= First Committee Draft , and TS= Technical Specification

Finally, CNC machines have to be able to translate the information defined by STEP-NC such as workingsteps to axis motion. STEP-NC has been formalised as two ISO standards namely ISO 14649 and ISO 10303-AP238 (Nassehi, 2007). STEP-NC extends the ISO 10303 STEP standards with the machining model in ISO 14649, adding geometric dimension and tolerance data for inspection, and the STEP model for the bi-directional information transfer between CAD/CAM/CNC systems. In the future, transferring files between CNC machines would be possible by considering the range of machine capability. However, CNC machines equipped with STEP-NC controllers are not commercially available at the moment.

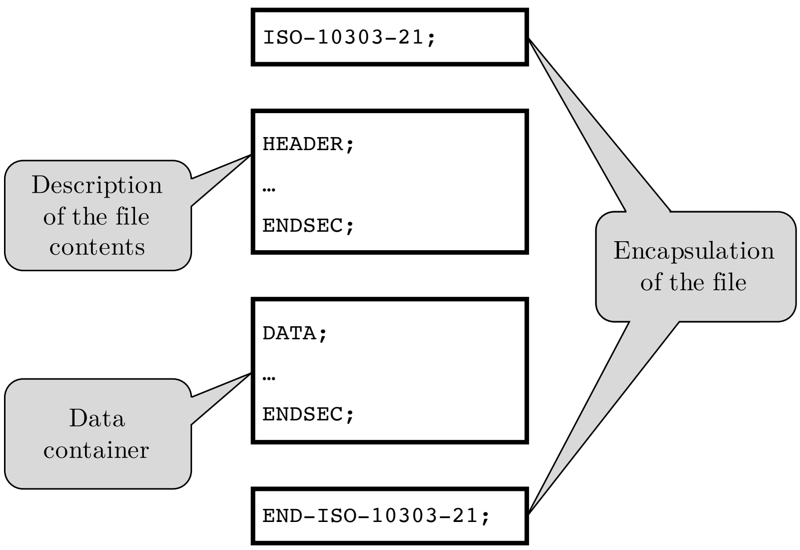


Figure 2-4: Structure of a part-21 file (Zhao et al., 2008)

Impacts of using STEP-NC in CAD/CAM/CNC chain are as follows:

CAD: STEP-NC includes AP238 and ISO 14649. AP238 contains design features, tolerances and ISO 14649 defines the working steps to machine a part. STEP-NC does not require CAM software. If the CAD system does not include CAM functionality, then AP-203 2, AP-214 and AP-224 will become main outputs for CAD.

CAM: STEP-NC allows CAM software to exchange data with other CAM systems.

CNC: STEP-NC gives more information about the product and process to the machine. This makes CNC machines more intelligent. Also, STEP-NC enables to send cutting data corrections such as readjusted feeds and speeds back to the designers and engineers.

Post processors: With STEP-NC it is encouraged that there is no need for post processors from CAD/CAM systems to generate G-code. The module used instead is a STEP-NC translator. This translator, converts the working steps in a STEP-NC file into the machining instructions understood by the CNC.

Process planning: The process planning based on STEP-NC standard contains the following transformations: A CAD system generates AP-203 2 or AP-224. A process planning system reads AP-224 and writes AP-240. A CAM system reads AP-240 and writes AP-238. A CNC system reads AP-238.

The abstract view of the above transformations has been depicted in Figure 2-5.

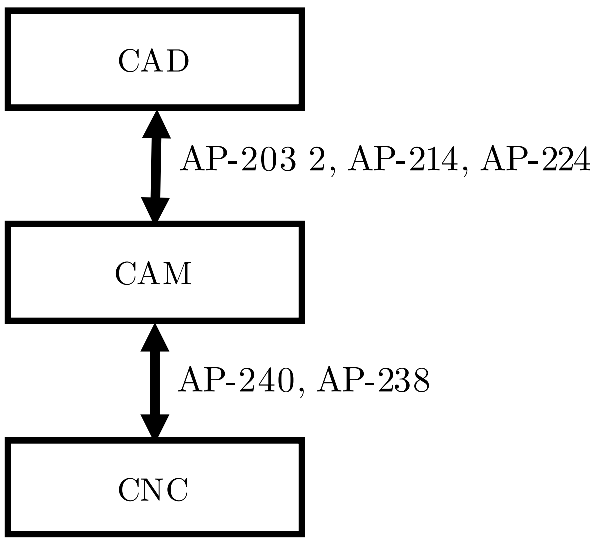


Figure 2-5: Information flow of STEP-NC in CAD/CAM/CNC chain

2.5 Manufacturability considerations in process planning

The task of process planning involves a series of activities that seek to define the necessary steps to change the shape of the raw stock to the desired product (ElMaraghy, 1993; Xu et al., 2011; Yusof and Latif, 2014).

The first step is the interpretation of the design which is usually achieved by the aid of a CAD system. The requirements of the products such as the raw material properties and the finished tolerances will be studied and interpreted. At this stage, the process planner will decide if it is possible to machine the part with the available resources. At present, this decision-making is totally based on the experience of the process planner. His knowledge about the available cutting tools and the condition of the machines on the shop-floor affects the process planning outcome.

The next step is to adjust the cutting feeds and speeds. These settings will change dependant on the material properties such as the stiffness and hardness of the material (Ridwan and Xu, 2013). Cutting feeds and speeds are normally suggested by cutting tool suppliers. These settings are collected by machining experiments on different materials and machines. Thus, the suggested cutting feeds and speeds may not always yield the expected result. This is mainly because of the external machining factors such as temperature and vibration. In addition to this, the cutters available in the shop-floor normally are not in a perfect condition (e.g. no worn or damaged teeth). Also, the cutting strategies (e.g. tool lead-in/lead-out) used to obtain the optimal cutting feeds and speeds by tool suppliers, may not be same as the cutting strategies used in a factory environment.

Finally, the manufacturing processes which can be used to form the part into its final shape have to be selected. This selection should be based on the available machines to accommodate the desired product requirements (Anjum et al., 2013). The available machining centres which can handle one or multiple machining processes have to be allocated. This allocation should be based on the capability of the machines such as available working volume, finished accuracy, available tools, maximum feed and etc. Currently, process planners utilise their knowledge of the technological capabilities of various types of machines to select the most suitable machine for a specific job (Wang, 2013). This knowledge can assist planners to generate reliable process plans based on the available manufacturing resources. In order to facilitate this decision-making process, it is essential to understand the machining resources through accessing machine tool models (Brecher et al., 2009).

2.5.1 Machine tool models

The result of process planning is highly influenced by functional and mechanical properties of selected machine tools. Information of functional properties, such as ability to perform different types of machining operations, is necessary in feature and operation sequencing (Chryssolouris, 2006). Machine tool models are conceptual representations of the real machine tool, and provide logical frameworks for representing its functionality in the manufacturing system (Brecher et al., 2009; Vichare et al., 2015). There are numbers of approaches to represent machine tool model (Kjellberg et al., 2009; Vichare et al., 2009). These approaches are categorised as follows:

- Graphical simulation: Physical machine elements can be modelled by accurate graphical representation. This graphical representation provides visual confirmation of machineability based on the kinematic reachability of the machine. Software packages such as machine tool builder in Siemens NX (Siemens NX, 2015), Vericut from CGTech (Vericut CGTech, 2017) and Delmia from Dassault Systemes (Delmia VNC, 2017) use graphical simulation to show kinematic relationships between the various physical elements of the machine. It is also possible to model auxiliary attachments of machine such as clamps, gantries, pallet changers and etc.

- Process capability representation: Transforming a workpiece from its original shape can be done by combinations of various technologies such as milling, drilling, turning and etc. This approach represents the manufacturing resources as a collection of process capabilities (Nassehi et al., 2012).

- Kinematic representations: Kinematic modelling of machine tools offers graph representation of the links between individual mechanic machine elements. Figure 2-6 shows the kinematic chain for a 5-axis machine tool. By following the kinematic chain from workpiece to tool it is possible to find the position and orientation of the tool in respect to machine origin. By calculating inverse kinematic, it is possible to find axes configuration, which requires complex mathematic computation. Similar modelling approach has been done previously for 5-axis machine tools by Suh et al. (2003b) and Yuen et al. (2013).

- Unified manufacturing resource model: Vichare et al. (2009) proposed a STEP-NC compliant Unified Manufacturing Resource Model (UMRM) for representing information regarding a variety of manufacturing resources in the decision-making process. The UMRM has been utilised to represent:

  - various configurations of conventional CNC machine tools including turning centres and multitasking turn-mill centres,

  - various auxiliary devices and material handling system configurations prevalent in the manufacturing industry,
  - a variety of fixturing devices used for enhancing the machine tool's workpiece holding capabilities.

- The STEP-NC standard machine tool data model is described in ISO 14649 as the Part 200 series. Part 201 on milling and turning is now a technical specification (ISO 14649-201, 2011). Rapid prototype (ISO 14649-17, 2017) and contour cutting (ISO 14649-15, 2017) machines have already been drafted to be standardised later as part of the series. The complete details of ISO 14649 standards has been given in Section 2.4.2.

	Legend
F	Machine Frame
1	Machine Bed
2	Rotary Table
3	Table
4	Slide
5	Column
6	Spindle
T	Tool
W	Workpiece

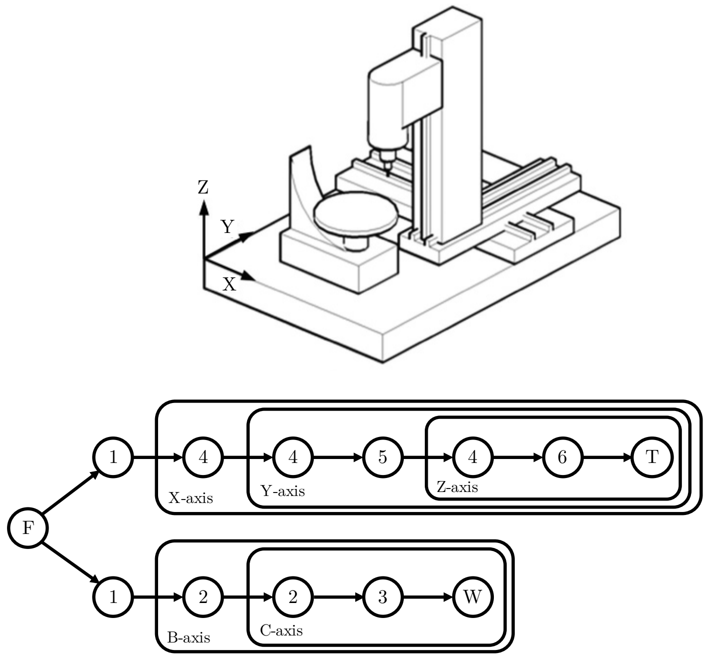


Figure 2-6: Kinematic chain representation of 5-axis machining centre (Nassehi and Vichare, 2009; Vichare et al., 2015)

- Other approaches: Suh et al. (2003b) proposed a machine tool model to represent the manufacturing elements and process mechanics in virtual environment. Tanaka et al. (2008) presented a semantic model of 5-axis machine centre based on its kinematic chain.

Various efforts have been made to develop a comprehensive machine tool model to be used in decision-making. As a part of NIST Rapid Response Manufacturing (RPM) research (Jurrens et al., 1995), information required by resource models has been classified as categories, attributes and relationships for development a common representation of manufacturing resource elements. A knowledge representation language known as "LOOM" (Wilczynski and Lipkis, 1993) has been used to model industrial CNC machine tool structures. However, use of LOOM remained controversial due to the high start-up cost and difficulty in handling semantic terms in machine tool modelling. Object-oriented manufacturing resource modelling (OOMRM) (Zhang et al., 1999) approach has been utilised for describing manufacturing resource capability and capacity in object-oriented manner. This research has been focused on the structural properties of the machine such as dimension, part position and tool orientation. Internet based virtual machine tools has been used (Suh et al., 2003b) to develop a prototype of web based virtual machine tools representing the geometric model of machine tool and kinematic movement of different elements. The machine tool model includes modules for defining the configuration of the overall machine tool structure, the geometric shape of mechanical units, and the kinematic relationship between mechanical units.

ISO 14649-201 (2011) has been published to represent machining resources and also other additional auxiliary devices such as coolant system, cutting tool changing mechanisms, probing system and bar feeder. Object-oriented structure uses ISO 14649-201 to classify mechanical resources to their application domains (e.g. material removing processes, material handling processes, assembly processes, material adding processes and measurement processes etc.). EXPRESS and EXPRESS-G languages are used to represent the relationships between these domains as it has enough portability, flexibility, and extensibility. The structural assembly information shown in Figure 2-7 can be utilised to define a machine tool for various application domains. Table 2.3 listed the entities used for a typical 3-axis CNC machine definition. Two different definitions for a machine tool in ISO 14649 part 201 are "`machine_tool_specification`" and "`machine_tool_requirements`". "`machine_tool_specification`" is used to define set of capabilities that can be offered by an existing machine tool. "`machine_tool_requirements`" is where the STEP-NC programmer wants to explicitly define what sort of a machine would be able to make a part. "`machine_tool_requirements`" is a set of requirements.

2.5.2 Machining errors

The accuracy of machining can be evaluated either under quasi-static or dynamic conditions. Quasi-static errors are those between the tool and the workpiece that are slowly varying with time and related to the structure of the machine tool itself. These sources include the geometric errors, errors due to dead weight of the machines' components and those due to thermally induced strains in the machine tool structure. Dynamic errors on the other hand are caused by sources such as spindle error motion, vibrations of the machine structure, controller errors etc. These are more dependent on the particular operating conditions of the machine. Quasi-static errors account for about 70 percent of the total error of the machine tool and as such, are a major focus of machining error in this research (Ramesh et al., 2000a).

Table 2.3: Definition of a machine tool (ISO 14649-201, 2011)

Entity	Data model description

Machine tool	`ENTITY machine_tool;`
	`SUPERTYPE OF (ONEOF(machine_tool_specification,`
	`machine_tool_requirements));`
	`description : text;`
	`END_ENTITY;`

Machine tool specification	`ENTITY machine_tool_specification`
	`SUBTYPE OF(machine_tool);`
	`machine_class : machine_class;`
	`device_id : device_id;`
	`machining_capabilities : SET [1:?]`
	`OF machining_capability;`
	`measuring_capability : OPTIONAL`
	`measuring_capability;`
	`location : OPTIONAL locator;`
	`installation : OPTIONAL installation;`
	`nc_controller_information : nc_controller;`
	`environment : OPTIONAL environmental_evaluation;`
	`its_elements : OPTIONAL SET [1:?]`
	`OF machine_tool_element;`
	`END_ENTITY;`

Machine tool requirements	`ENTITY machine_tool_requirements`
	`SUBTYPE OF(machine_tool);`
	`number_of_tools_in_tool_magazine :`
	`OPTIONAL count_measure;`
	`machining : SET [1:?] OF`
	`machining_capability;`
	`spindles : OPTIONAL SET [1:?]`
	`OF spindle_capability;`
	`positioning : OPTIONAL`
	`positioning_capability;`
	`axis : OPTIONAL axis_capability;`
	`touch_probing : OPTIONAL measuring_capability;`
	`automatically_pallet_changeable : BOOLEAN;`
	`END_ENTITY;`

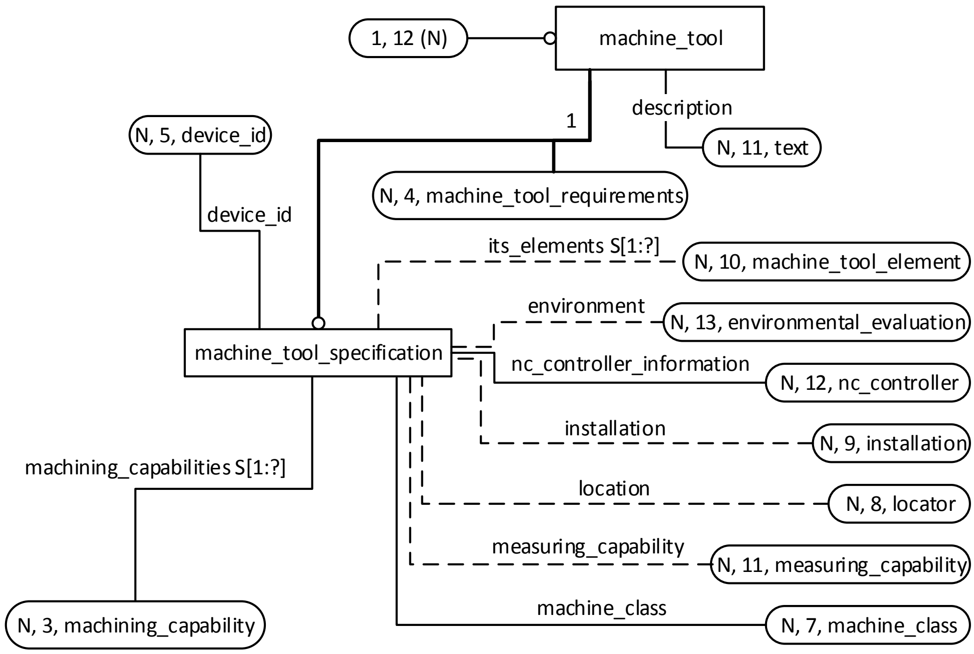


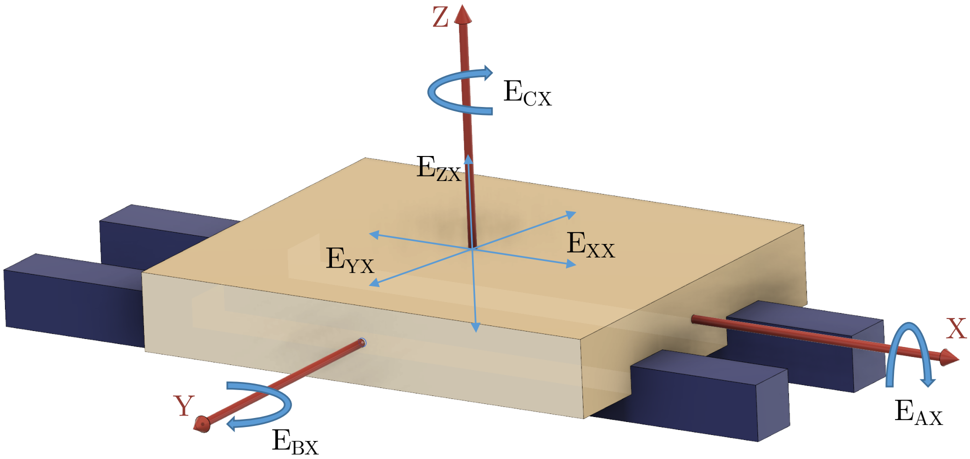
Figure 2-7: EXPRESS-G representation of a machine tool (ISO 14649-201, 2011)

Errors which affect machining accuracy can be grouped into three major classes namely (I) geometric errors, (II) process errors and (III) thermal errors (Ramesh et al., 2000a).

2.5.2.1 Geometric errors

Geometric errors are those errors that exist in a machine on account of its basic design, the inaccuracies built-in during assembly and as a result of the mechanical components used on the machine (Fu et al., 2015). This type of error happens when machine tool motion structures do not lie perfectly true and deform due to gravity and alignment. The assembly of mechanical components which maintain a relative position between specific objects may deform. This will lead to the nominal location of end-effector not corresponding with the actual location of it in the space. Using jigs and fixtures through the assembly process of the motion platform can also minimise the risk of having this type of error. Geometric errors are concerned with the relative motion errors of several moving machine components that need to move to position tool tip (Zhang et al., 2013). This error is comprised of straightness error - in how true to a linear axis each guide runs; squareness error - in how far from 90°each axis lies from each other; and rotational error - in how each axis is aligned in terms of yaw, pitch and roll (Ibaraki et al., 2010).

ISO 230 defined six errors for a vertical X movement including the positioning error, two straightness error motions, roll error motion and two tilt error motions. These errors have been depicted in Figure 2-8.



E_{AX}: angular error motion around A-axis
E_{BX}: angular error motion around B-axis
E_{CX}: angular error motion around C-axis
E_{XX}: linear positioning error
E_{YX}: straightness of X in Y direction
E_{ZX}: straightness of X in Z direction

Figure 2-8: Angular and linear error motions of a component to move along a straight-line (ISO 230-1, 2012)

Geometric errors form one of the biggest sources of inaccuracy (Stephenson and Agapiou, 2016). Also, geometric errors are repeatable, stable and measurable (Fu et al., 2015). Thus, they are considered as the main indicators of a machine tools' health (Parkinson et al., 2012). The geometric error data utilised in this research is adapted from ISO 230 (ISO 230-1, 2012). Next section reviews other standards released so far for the health assessment of machine tools.

2.5.2.2 Process errors

The act of processing a part will not be fully consistent on part-by-part basis. The error which is forced into the machining of a part by external factors in this manner can be defined as a process error (Agapiou et al., 2016). Process errors are classified as dynamic errors, as they change each time a part is machined. For instance, cutting tool which may wear over time, can affect the finished size of products. This can be compensated manually or automatically by adjusting tool diameter offset in machine controller (Lei et al., 2014).

Using correct machining feed and speed can reduce the risk of having process errors. Excessive machining feed can lead to rapid tool degradation as well as poor finish (Khoshdarregi et al., 2014). Material properties can affect the accuracy of machining.

This can be seen especially in compound materials such as alloys, composites and extruded materials. In this case the density of the material may change across the body of material and therefore need different cutting forces during machining.

Excessive cutting force imposed by the machine tool structure is classified as process error. Minimising the cutting force generated by the dynamic stiffness of the machine tool can reduce the risk of having this type of error. Available models to predict cutting force error can be used dynamically to compensate this error (Kaymakci et al., 2012).

Fixturing error can happen by small variations of workpiece clamping location on the machine bed. Normally, workpiece positioned on the machine bed using a vice or clamp. Inaccuracy due to poor fixturing may happen especially in high speed machining. Machining accuracy is tightly linked with the overall rigidity of the machine structure and the positional accuracy of workpiece.

Process errors may occur in machining in the combined effects of the aforementioned errors. For example, excessive cutting force may cause fixturing errors. Equally, cutting force error may generate excessive tool vibration. This may contribute to tool wear and eventually break cutting tool.

2.5.2.3 Thermal errors

Thermal errors are those that cause small variations in the distance between workpiece and end-effector due to deformation and expansion of the machine elements which are caused by heat. Relative movement between the different components of the machine generates heat at contact zones and this heat results in the expansion of the machine elements (Ramesh et al., 2000b). Up to 70% of the machining inaccuracy may be caused by thermal deformation (Jedrzejewski et al., 2008). This error may occur due to thermal expansion of the machine structure, workpiece and tool (Bryan, 1990).

Ideally, machine tools need to be kept in a controlled ambient temperature (Ramesh et al., 2000b). There are two types of thermal errors in general: The first type is the thermal errors that changes with the change of temperature but not the axis position. The second type deals with the thermal errors which can cause deflections in the linear axes. The effect of the second type error is more significant. There are numbers of solutions to reduce the thermal errors such as control of heat flow into the machine tool environment, redesign of the machine tool system to reduce sensitivity to heat flow and compensation through controlled movement (Ramesh et al., 2000b).
Ramesh et al. (2000b) investigated the thermal errors through three stages: modelling, measurement and compensation and reviewed each of them in detail. In high volume

production, the majority of the thermal errors come from the roughing operations where the rate of stock removal is substantially higher than finish cutting. The magnitude of the total error caused by heat can be estimated by knowing the thermal expansion coefficient of the materials (Yang et al., 1999). This has been investigated on the machine ball screw and the machine guideway structures. The results show that the ball screw only affects linear positioning errors, whereas the guide way causes angular errors as well as linear positioning errors (Yun et al., 1999).

2.5.3 Machine tool testing and verification standards

Standards which have been published on machining accuracy under quasi-static mode have been outlined (ASME B5.54, 2005; ISO 230-1, 2012). Most of the machine tool testing and verification standards are focusing on the following areas: (i) Elements of machine tool errors, (ii) Methods and test procedures for identifying machine tool errors, (iii) A methodology to evaluate machine tool positional accuracy and repeatability, and (iv) Guidelines for representing health parameters.

Although a number of standards and guidelines now exist outlining how to evaluate machine tool positional accuracy and repeatability such as ISO 230 (ISO 230-1, 2012), VDI/DQG 3441 (VDI/DGQ 3441, 1977), JIS B6330 (JIS B6330, 1980) and GB/T 17421 (GB/T 17421, 2003), they differ in their analysis procedures and in key parameter definition.

Aforementioned standards have been successfully used to compare performance of different machine tools in the past. To clarify the methods of performance evaluation of new and reconditioned machines, existing standards on machine tool accuracy has been outlined. Then, the relevant testing procedures which reflect machine tool health has been selected. These testing parameters have been used to build machining capability profile data model. Developed data model, can be processed to generate periodic performance checks for any type of machine.

2.5.4 Kinematic models for machine tool accuracy

Kinematics is “a branch of mechanics that describes the motion of points, bodies (objects), and systems of bodies (groups of objects) without considering the mass of each or the forces that caused the motion” (Whittaker, 1988). In kinematic modelling, mathematical models of kinematic quantities such as joints and links describing the movement of the machine tool tip are developed. Kinematic calculations normally start with determining the number of joints and links. Next, transformation matrices are used to quantify the displacements through the kinematic chain of a machine tool. These transformations have been extensively represented by Denavit and Hartenberg (1964).

The Denavit and Hartenberg notation relates the position of local reference systems associated to each element of a spatial manipulator to find the position of the tool grasped by the manipulator. Denavit and Hartenberg is a systematic way to find the position of the end-milling tool and has been specially developed for the study of complex spatial manipulators such as three-axis, five-axis or parallel kinematic machines. In machine tools containing only linear axes, the process of finding the tool tip position is simple because the orientation of the cutter remains unchanged. In post processing for machine tools containing rotational axes, the calculation is more complex. This is because of the varying orientations of the cutters with respect to the workpiece, the coordinate transformation matrix must be calculated to obtain the tool tip position.

The global transformation matrix $T_{4\times4}$ is obtained by the product of the transformations between successive coordinate systems associated to the mechanism's elements, from the absolute (X_0, Y_0, Z_0) reference system to the tool centre point (t) system (X_t, Y_t, Z_t). Thus, this matrix is a function of the geometrical parameters of the mechanism's elements and the position of the machine axes. Each scheme of kinematics and each machine presents its own transformation matrix $\mathrm{T}_{4\times4}$. In the Denavit and Hartenberg formulation, the origin of the coordinate system is associated with the ith element and is located on its joint with the element $i+1$. The conversion from the $(i-1)$th system to the ith system is performed through a number of transformations. For example, the transformation matrix for a typical five-axis machine shown in Figure 2-9 can be ascertained from the following transformations (Lamikiz et al., 2008):

1. A rotation θ_i about the Z_{i-1} axis to bring X_{i-1} parallel with X_i
2. A translation d_i along the Z_{i-1} axis to make the X axes collinear; this is known as the distance between elements
3. A translation a_i along the X_i axis to make the Z axes coincide
4. A rotation α_i about the X_i axis to bring Z_i parallel and coincident with Z_{i+1}

The Denavit and Hartenberg matrix A_i for the above individual transformation can be obtained as follow:

$$\begin{pmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

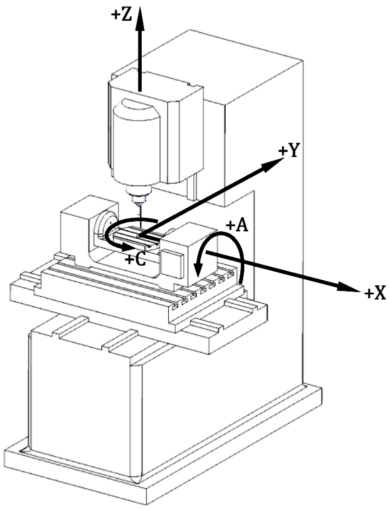


Figure 2-9: Geometrical model of a typical five-axis machine (Flynn et al., 2015)

Global transformation matrix for the series of axes rotations and translations ($A_1 ... A_5$) can be obtained by Equation 2.1. The relative position and orientation of the tool coordinate system (A_t) with respect to the workpiece coordinate system (A_w) is expressed by this matrix.

Equation 2.2 determines the coordinates of a vector in the absolute coordinate system ($\{X\}$) which can be obtained from the multiplication of that vector inside the machine working volume ($\{X_m\}$) and the global transformation matrix (T).

$$T = A_w \times A_0 \times A_1 \times A_2 \times A_3 \times A_4 \times A_5 \times A_t \tag{2.1}$$

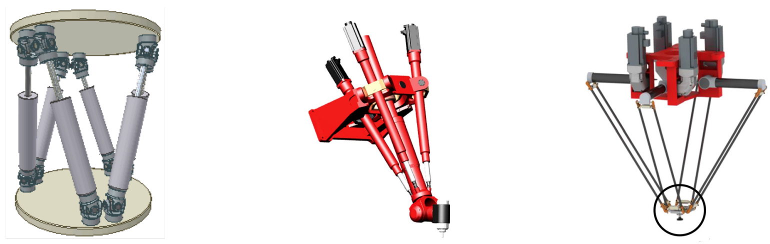
$$\{X\} = T \times \{X_m\} \tag{2.2}$$

Due to the machine kinematic errors, the result vector ($\{X\}$) does not necessarily represent the actual displacement of the tool tip. This has been investigated by Flynn et al. (2015) to measure the tilt error in a five-axis machine tool.

Robots are defined as "automatically controlled, reprogrammable multi-purpose manipulators, programmable in three or more axes, which can be either fixed in place or mobile for use in industrial automation applications" (ISO 8373, 2012). Serial manipulators and parallel manipulators are the two most popular types of robots that offer dissimilar kinematic capabilities. The most commonly used robots for the industrial applications are serial kinematic robots. These robots consist of the end-effector and the base which are connected by a sequence of links. On the other hand, parallel kinematic robots, are formed by closed kinematic chains. In these machines the end-effector is connected to the base by means of several kinematic chains (Merlet, 2006).

In contrast to serial kinematic machines, parallel manipulators often have complex kinematic relationships. This complexity is largely due to the closed-loop nature of the machine structure, where multiple kinematic chains connect to a common end-effector plate. Therefore, in order to position the end-effector, there must be compliance between each of the kinematic chains that connect the base to the end-effector. Parallel kinematic machines (PKM) were originally introduced by Gough and Whitehall (1962). PKM was used for tire testing and flight simulation for the first time by Stewart (1965). Since parallel architecture of PKMs help with the flexibility and structural rigidity of these machines, they have been subject of great interest in industry. Some of the most extensively utilised parallel manipulator configurations to date include the 'Stewart' platform, the 'Tricept' robot and the 'Delta' robot (Weck and Staimer, 2002b); all of which are presented in Figure 2-10. PKMs normally classified by the degree of freedom (DOF) of the end-effector and the actuator and joint arrangement. The number of DOFs of a parallel mechanism can be any number between two and six. For different kinematic configurations of PKMs, the reader is directed to the book published by Liu and Wang (2014).

The accuracy of PKM machines have been identified as a critical point which have not yet reached industrial application level (Marquet et al., 2002). Thus, the kinematic error of these machines have been considered as the major contributor to the overall accuracy of PKMs (Weck and Staimer, 2002a).



(a) (b) (c)

Figure 2-10: Different kinematic structure of PKMs: (a) Stewart platform (b) Tricept robot (c) Delta robot (Weck and Staimer, 2002b)

2.5.5 Machine tool health capturing techniques

A healthy machine tool is defined as a machine tool that its performance has been checked though a series of tests against the machine tool specifications. Since the introduction of the machine tool capability profiles by Newman and Nassehi (2009), several machine tool modelling techniques have been proposed to capture and store the condition of machines. All of these researches are focused on representing machine tool components using data models. Vichare et al. (2015) used STEP structure to model machine tool health, this model has been presented using ISO 14649-201 (2011) to model a machine tool. The data model presented in Vichare et al. (2015) has been incorporated with machine tool testing standards such as ISO 230-1 (2012). The methodology stated in (Fesperman et al., 2015) has been used to generate a profile of machine. The geometric errors of the machine have been saved in the format of eXtensible Markup Language (XML). The proposed methodology has been used to evaluate the performance of a machine virtually. A formal ontology approach for presenting machine tool health condition has been used in Sadeghi and Farhad (2013). The ontological reasoning used in this research has been tried in macro process planning stage.

2.6 A critique of the process planning approaches

Researchers are focused on optimising process planning system to achieve the following goals (Alting and Zhang, 1989):

- Generating process plans without human intervention (Xu et al., 2011)
- Updating existing process plans based on available information on machines, cutting tools, feeds and speeds (Ridwan and Xu, 2013).
- Capturing the knowledge of the skilled process planners to automate plans for group of components (Wang, 2013).
- Optimising machining time and cost (Chryssolouris, 2006)

Their efforts have undoubtedly achieved certain level of success; however, few CAPP systems have gained industry acceptance. This could be attributed to the following shortcomings in terms of problem modelling:

1. Most reported CAPP systems are designed based on the assumption that machine tools work on their best ability. However, there are huge numbers of factors which influence on machining capability such as kinematic errors, geometric errors and thermal errors.

2. Machine tool models used by CAPP systems do not represent machine tool health efficiently. For example, kinematic model of machine tools defined in ISO 10303 part 105 (ISO 10303-105, 1996), only focuses on the geometric representation of kinematic chain, and there are limited applications on the provided model. Therefore, decision-making based on these machine tool models is not reliable.

3. Decision-making techniques used in the reported literature (Alting and Zhang,

1989; Li et al., 2010b; Xu et al., 2011) are not capable to accommodate various machining parameters. The reasoning method used in these CAPP systems do not support machining complexities and thus the results can not be trustworthy.

In previous years major efforts have been made to design an expert system for manufacturing process planning such as intelligent tool selection (Fernandes and Raja, 2000; Lim et al., 2001), automated setup planning (Choi and Xirouchakis, 2014; Hebbal and Mehta, 2008) and computer aided operations selection and sequencing (Gen and Lin, 2014; Krishna and Rao, 2006). In reality, machine tools are not working on their ideal condition, and most of the time they fail to deliver quality parts. Errors may affect on the finished part quality are geometric errors, kinematic errors, thermal errors, cutting force induced errors and fixturing errors (Khodaygan, 2014; Ramesh et al., 2000a). There errors effect on machine tool health over its operational life span, and cause lots of variations in delivered tolerances.

With the complexity that exists in the CAD/CAPP/CAM chain for delivering efficient and well-structured process plans, the CAPP function has been limited most of the time. Also, integrated CAD, CAM and CAPP software such as Catia, NX, Pro/Engineer and Inventor has not effectively satisfied industry so far (Vichare et al., 2009, 2015). The ultimate goal in CAPP is the ability to automatically generate production plans for new products, or dynamically update production plans on the basis of resource availability (Horvath, 1996). Although, machining resources availability and capability have to be considered through entire CAD/CAPP/CAM chain, however, lack of having this knowledge can deliver poor and unreliable process plans.

The recently developed standard which addresses this issue is ISO 14649 part 201 (ISO 14649-201, 2011). Two separate data models have been introduced to describe machine tools. '`machine_tool_specification`' and '`machine_tool_requirements`' data models cover milling machines, drilling machines, turning machines and multi-tasking machines. '`machine_tool_specification`' is used to define set of capabilities that can be offered by an existing machine tool. '`machine_tool_requirements`' is where the STEP-NC programmer wants to explicitly define what sort of a machine would be able to make a part. Both data models extend '`machining_capability`' and '`machining_capability_profile`'. However, the applications of these data models are limited to a few capability parameters such as numbers of tools, maximum spindle rotation and etc. Also, there is no considerations of machine ageing in both following definitions:

```
ENTITY machining_capability;
capability :  machine_capability_profile;
machining_accuracy :  OPTIONAL text;
description :  OPTIONAL text;
machining_size :  OPTIONAL machining_size;
```

```
END_ENTITY;


TYPE machining_capability_profile = ENUMERATION OF
(BORING_CAPABILITY, DRILLING_CAPABILITY,
GUNDRILL_CAPABILITY, MILLING_CAPABILITY, TURNING_CAPABILITY);
END_ENTITY;
```

2.7 Research gaps

Careful consideration of the literature presented in this chapter leads to the identification of the following gaps in the state-of-the-art:

(i). There is no available machine tool model for capturing actual capability of manufacturing resources. The available machine tool models only provide the information about the nominal state of the machines to process planning systems and capturing the actual status of machine tools has not been a focus of the reviewed literature.

(ii). There are no approved CAPP systems which consider process capabilities such as tolerance and surface finish throughout a machine life as it degrades.

(iii). Capability profiling or capture of historical health has been addressed in other domains such as healthcare (Wang et al., 2018), robotics (Seiger et al., 2015) and warranty (Humphreys et al., 2002). In the area of machining, research has been done to capture actual machine capability. These works have been reviewed in Section 2.5.5. Most of these studies are focused on representing machine tool conditions using data models. However, a validated integrated framework for process planning based on actual machine status is still missing.

This research endeavours to fill this gap by proposing and validating the CAPPable framework as presented in Chapter 3 onwards.

With the extensible nature of STEP and availability of the EXPRESS data modelling language, the STEP-NC standard can be augmented to develop a machining capability profile. The author believes that this approach would be appropriate for development of the CAPPable framework because:

- STEP-NC standard provides process planner with machining processes to manufacture a part. Provided machining tasks in STEP-NC can be used to develop a capability profile since machining operations are known at this stage.

- STEP-NC contains product requirements in data model structure which can be used for reasoning purposes. The data model is also computer understandable to assess the capability of machine.

- STEP-NC extends the ISO 10303 STEP standards with the machining working steps in ISO 14649, adding geometric dimension and tolerance data for inspection. Providing tolerances by STEP-NC, it is possible to compare the machining requirements with available machining resources.

Chapter 3

Specification of a capability based process planning framework

3.1 Introduction

This chapter introduces the reader to the overall concepts and specifications for the research. It is divided into the design of the CAPPable framework, manufacturing capability profile descriptions and machine tool models. Process plans are generated based on the nominal capability of manufacturing resources can result in the rejected parts. This is due to fact that, manufacturing resources degrade over their operational life. Nominal resource models are not capable to express this degradation since they are not based on the actual manufacturing resource status. This chapter will describe the theories and methodologies that were used to achieve the research goals and the objectives defined in Chapter 1.

3.2 Overview of capability based process planning

At present, process planners utilise their knowledge of technological capability of various types of machines to select the most suitable machine for a specific job. Figure 3-1 presents the current process planning system without considering the capability of machine tools.

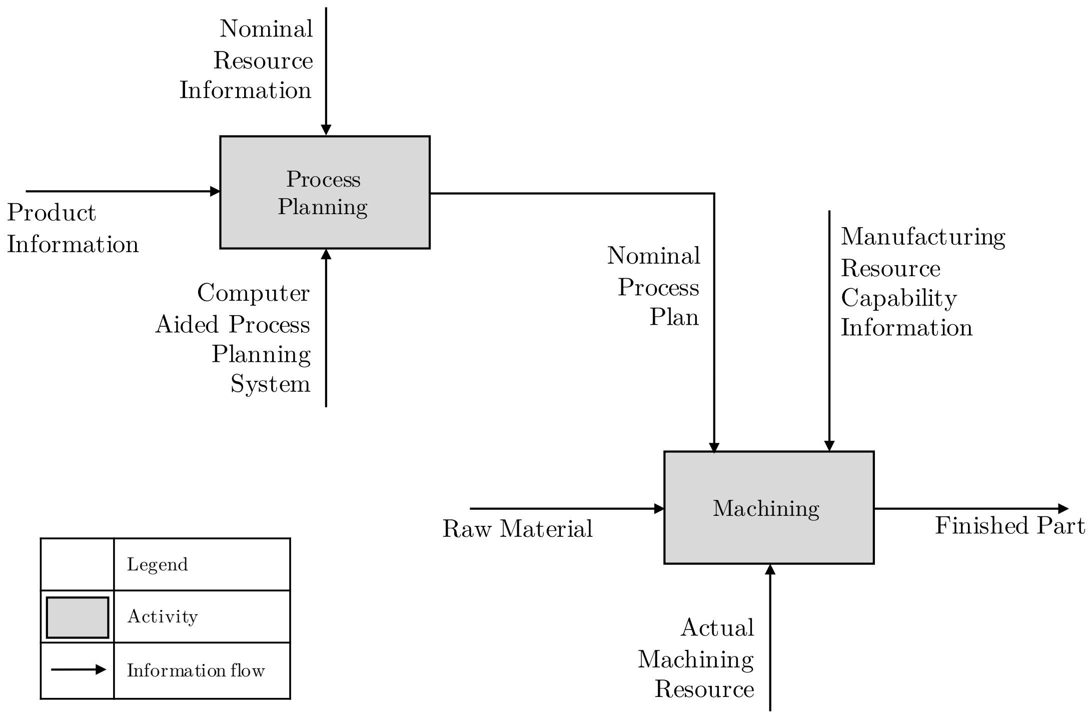


Figure 3-1: Functional view of process planning adapted from Newman and Nassehi (2009)

Although as reported in Chapter 2 significant research has been undertaken in the area of CAPP systems, still these systems are not fully automatic. In order to make the process fully automatic or to provide better information to the process planner who oversees the function of the semi-automatic decision support system, the actual status of machining resources need to be added to the current CAPP systems. This has been identified as a requirement on the available CAPP systems (ElMaraghy, 2007; Li et al., 2010b; Xu et al., 2011).

In order to enable the process planning system to determine the most effective plan with respect to the actual available resources, it is necessary to provide resource information that reflects the status of the physical devices at the time that they will be utilised for manufacturing the part. This time-sensitive image of the resource, called by Newman and Nassehi (2009) a capability profile, is a representation of the capabilities that a specific machine tool will be able to provide in a specific time on a particular product. Figure 3-2 illustrates how the existence of the capability profile influences process planning based on actual resource information and provides the overall framework for a Computer Aided Process Planning system based on actual machine capability (CAPPable). The capability profile can be generated by many techniques including prediction and online monitoring of resources and is subjected to policies set by the user. Regardless of the generation technique, it is imperative that the profile is associated to the representation of the resource within the manufacturing information. The CAPPable framework can then utilise this resource capability profile instead of the nominal resource information to create a capability adjusted process plan.

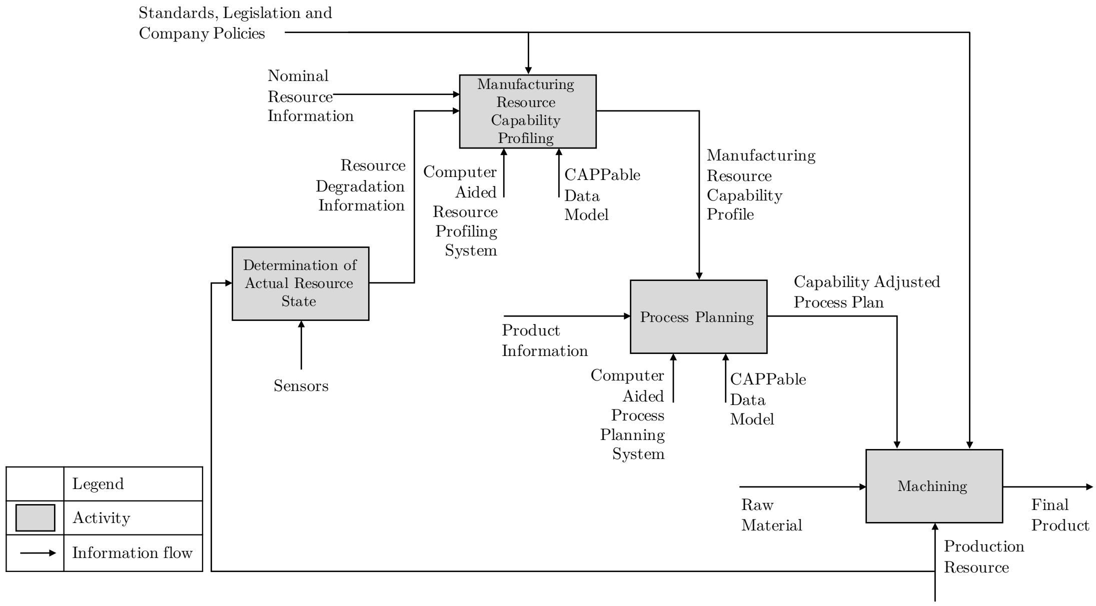


Figure 3-2: The functional model of process planning based on manufacturing capability profiles (Newman and Nassehi, 2009)

Strong reliance over the knowledge of operators to select appropriate machine tools based on their experience is now considered obsolete in current CNC manufacturing (Azab, 2016). This manual approach reduces the repeatability and the effectiveness of a manufacturing company in realising the full production potential (Stock and Seliger, 2016). So, there is a major requirement to seek standardised methods to represent manufacturing resource health.

This recent trend of automation and data exchange in manufacturing technologies has been introduced by Industry 4.0 with the goal of intelligent, resilient and self-adaptable machines (Kagermann et al., 2013). To accommodate this, machine tools are required to become aware of other manufacturing resources within the factory environment. Interconnected sensors and controllers can be used to support this communication (Lee et al., 2016). Using these sensors, machine tool data can be stored, accessed and exchanged across the different factories (Wang et al., 2016). Based on Industry 4.0, in smart factories, cyber-physical systems control physical processes, create a virtual copy of the physical machines and make decentralised decisions throughout the supply chain. A cyber-physical system (Baheti and Gill, 2011) is defined as transformative technologies for managing interconnected systems between its physical assets and computational capabilities.

Product models are well established commercially at the design phase with information relating to geometry, tolerances, functional capability, assemblies such as ISO 10303-

1 (1994) and ISO 14649-1 (2003). However, for developing a machining capability profile there is a need for machining resource data models, which can be utilised to represent machine tool functionality and consequential process capabilities for allocating resources for process planning and machining. This has been partially fulfilled by the emergence of ISO 14649-201 (2011), but additional machine capability data models have to be developed for the full assessment of the machineability of a part. As a result of this decision, the additional aspects of the data model have been developed using the EXPRESS data modelling language and are presented using the standard graphical representation of EXPRESS known as EXPRESS-G (ISO 10303-11, 1994).

3.3 EXPRESS

ISO 10303-11 (1994) is the international standard which defines the formal language known as EXPRESS to describe STEP-data requirements. EXPRESS is used to define data entities, attributes, inheritance, relationships, rules and constraints. It defines entities, application's objects, through focusing on its properties and constraints of an application domain. Object oriented language in EXPRESS allows users to define data types and object instances. EXPRESS can represent simple, aggregation, named, constructed and generalised data types. Table 3.1 lists the major categories of EXPRESS data types and the included data types in each category. The description of each category has been given in Section 3.5.1.

Table 3.1: EXPRESS data types classifications

Simple data types	Aggregation data types	Named data types	Constructed data types
Number	Set	Entity	Select
Integer	List	TYPE	Enumeration
Real	Array		
Logical	Bag		
Boolean			
String			
Binary			

The modelling mechanism in EXPRESS allows to define supertype/subtype inheritance relations. In addition, it allows for specifying an instance of supertype to be one of its subtypes, or to be of more than one of its subtypes using ANDOR or AND constraints. Functions within EXPRESS can operate a defined algorithm on its parameters to produce a single result value of a specific data type. EXPRESS also includes some built-in functions to evaluate mathematical expressions. Expressions can also be specified within EXPRESS; they are a combination of operators, operand and function calls to evaluate a value. There are arithmetic, logical, relational, membership and other operators that are defined to assist and extend the described EXPRESS capabilities as a data-requirement specification language.

EXPRESS-G, presented in ISO 10303-11 (1994), is a STEP graphical tool that aids the understanding of modelled data requirements using EXPRESS. Although EXPRESS-G can represent all data requirements, it does not have a defined way to represent modelled rules and constraints involved within an EXPRESS data model. Figure 3-3 shows the symbols used in EXPRESS-G to represent different data types and entity relationships defined in the EXPRESS model. Figure 3-4 uses an example of data requirements represented in EXPRESS-G and its related EXPRESS descriptions are as follows:

```
TYPE hairType = ENUMERATION OF (blonde, brown,
black, red, white);
END_TYPE;

ENTITY person
ABSTRACT SUPERTYPE OF(ONEOF(female, male));
firstName :  STRING;
lastName :  STRING;
nickname :  OPTIONAL STRING;
birthDate :  date;
children :  SET [0:?]  OF person;
hair :  hairType ;
DERIVE
age :  INTEGER := years (birthDate);
INVERSE
parents :  SET [0:2] OF person FOR children;
END_ENTITY;

ENTITY female SUBTYPE OF (person);
INVERSE
husband :  SET [0:1] OF male FOR wife;
END_ENTITY;

ENTITY male SUBTYPE OF (person);
wife :  OPTIONAL female;
END_ENTITY;
```

The `person` entity represented in this example can be one of two subtypes; they are `male` and `female` entities. The `person` entity has four mandatory but two optional attributes. Hair type, birth date, first name and last name are a must attributes; age attribute is derived, while nickname and children are optional attributes, as some persons may not have a nickname or children. The figure indicates that a person has one first name , last name, hair type and optionally a nickname while for a person having children, this could be from one to many children. The male entity could have a wife attribute that is represented by a female entity; this relation is inverted such that the female entity has a husband attribute that is represented as a male entity.

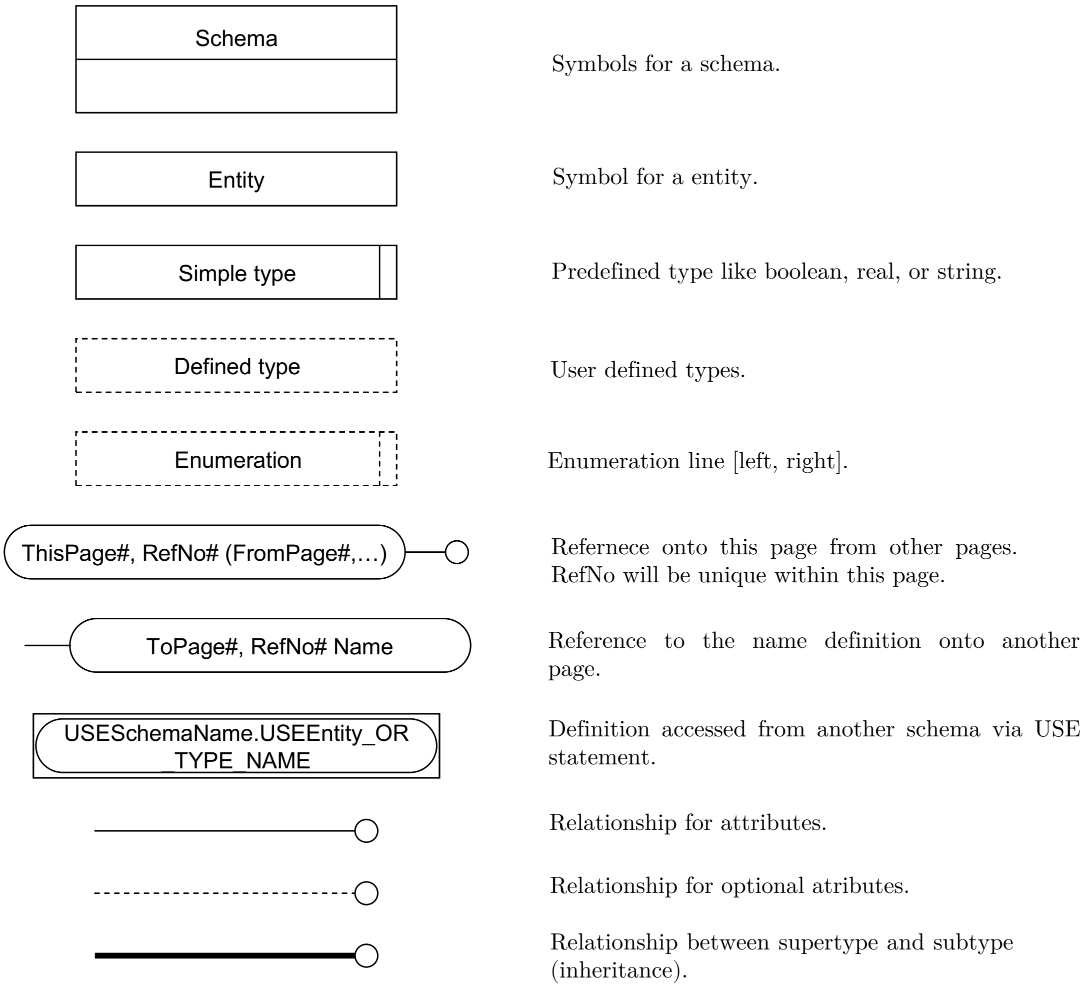


Figure 3-3: EXPRESS-G types and entity relations (ISO 14649-201, 2011)

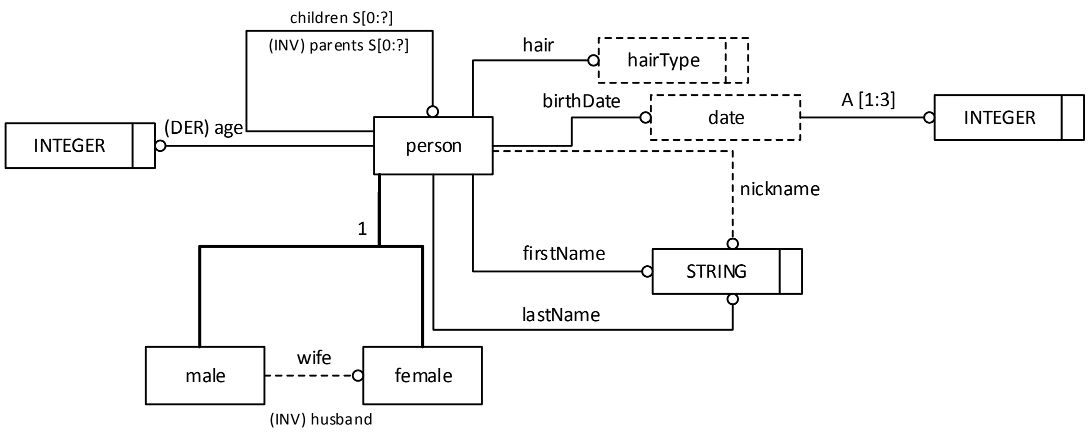


Figure 3-4: EXPRESS and EXPRESS-G illustrative example

3.4 CAPPable data model

Machine tools are degrading over their operational life, and the data related to this degradation needs to be stored on the presented framework. For this reason, the database '`machining_capability_data_point`' has been introduced. This database stores historical health data of a machine tool. '`machining_capability_data_point`' stores the machine tool health data collected from measurement tools such as a touch probe or a laser interferometer. Thus, it is essential to determine the factors affecting the machining results throughout the use of the CAPPable. This has been investigated in Section 2.5.2. Next, the machine tool health database is integrated with the machine tool capability profile defined in top level. The proposed framework can be used to evaluate the performance of different machine tools.
'`machining_capability_profile`' has been introduced by the author so that any machine tools with different capabilities can be added to the above definition. The capability data model developed for the proposed system, extends ISO 14649 part 201, and it is fully compatible with other STEP-NC standards. This has been illustrated in Figure 3-5. The developed data models have been surrounded with the red box, and the extended ISO 14649 part 201 data models have been surrounded with the dashed red box in Figure 3-5.
The main advantage of the aforementioned system, is that it enables process planner to make decisions based on the actual condition of the machining resources. Also, CAPPable provides process planners to work based on the design requirements. The extended STEP codes include dimensions and tolerances required for machining a part. CAPPable is an extension to STEP and STEP-NC platforms. So, it uses the information from design stage to decide whether it is possible to deliver the tolerances defined in STEP. This decision-making can be based on STEP-NC code which contains machining working steps and operations required to finish a job. A degraded machine tool can still be suitable for some jobs depending on the required tolerances. CAPPable can decide whether a degraded machine can meet design specifications considering design tolerances and dimensions. The following entities added to the data model description defined in ISO 14649 part 201 to store capability profiles:

```
ENTITY machining_capability_profile;
its_machine_tool : machine_tool;
its_data_points : SET [0:?] OF machining_capability_data_point;
END_ENTITY;

ENTITY machining_capability_data_point;
its_id : label;
its_machining_capability : machining_capability;
its_date : calendar_date;
its_time : local_time;
END_ENTITY;
```

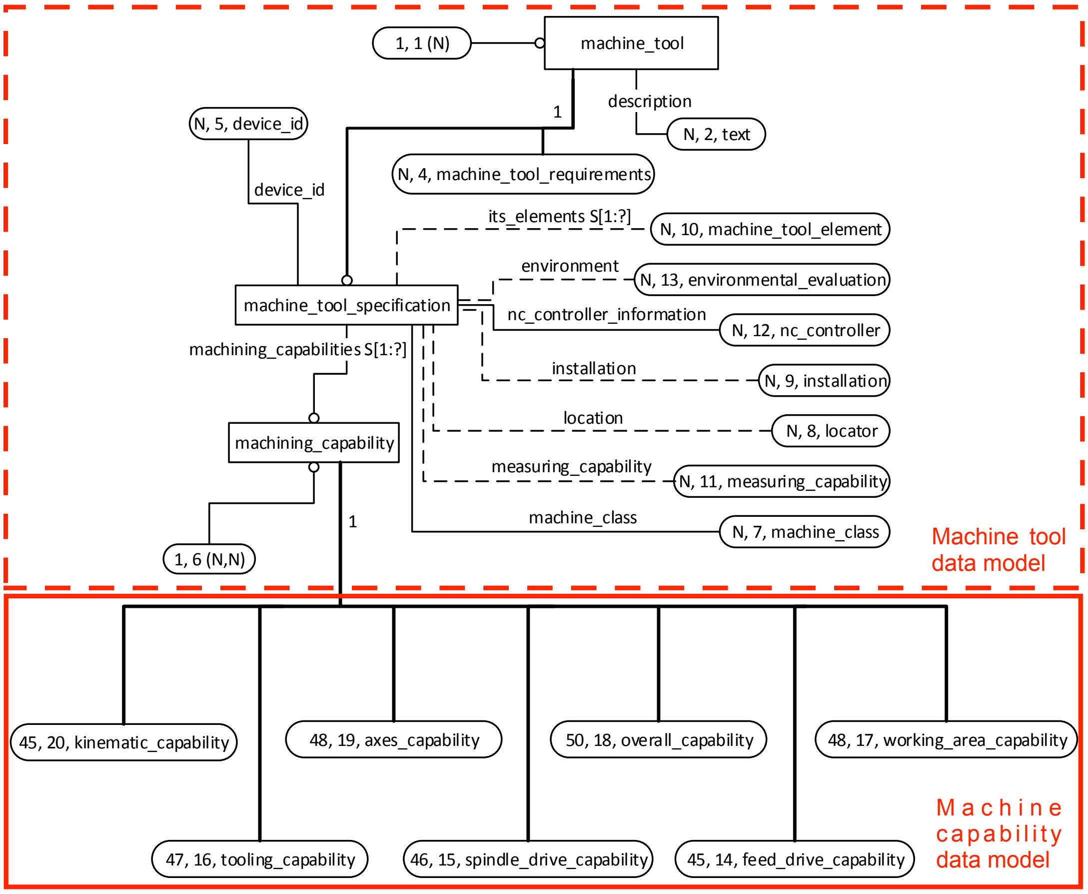


Figure 3-5: Connection of the proposed data model to ISO 14649-201

The top level structure of the machining capability profile data model is presented in Figure 3-6. From this figure a machineability capability profile consists of:

- Kinematic capability
- Feed and spindle drives capabilities
- Tooling capability
- Axes capability
- Working area capability
- Machine overall capability

This CAPPable data model has been designed to add the machine tool health data to the machine tool data model defined by ISO 14649 Part 201 (ISO 14649-201, 2011). The CAPPable data model can be fully integrated with other parts of the STEP-NC standards. A machining capability profile contains following machine tool health elements:

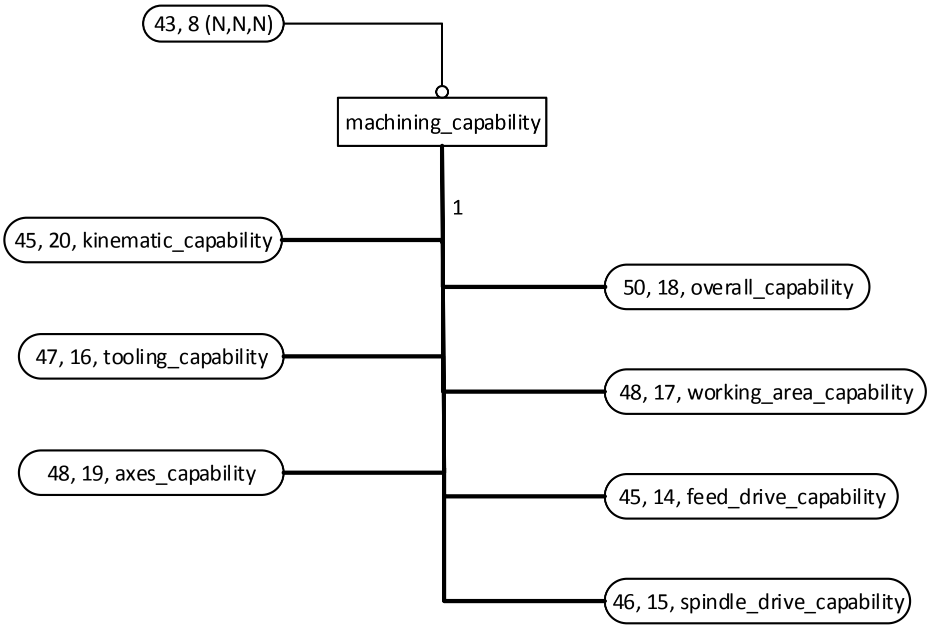
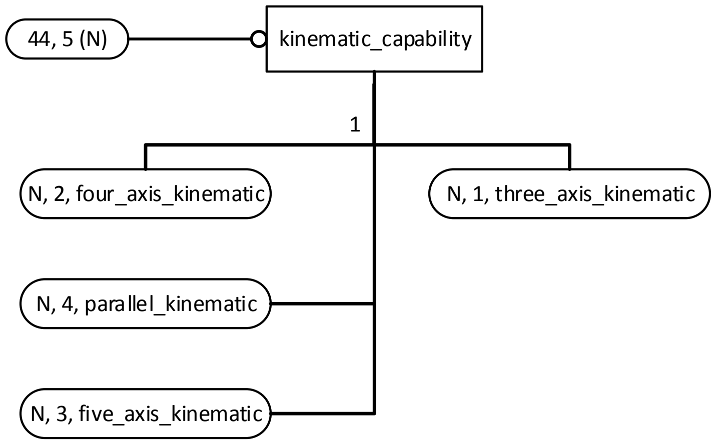


Figure 3-6: Express-G diagram of the proposed data model for machining capability profile

(i). Kinematic capability: Machine tools with different designs can locate their axes in different ways to machine the same part because they have different kinematic configurations. Because of kinematic arrangement, it is necessary to study the effects of various machine kinematic configurations on the movement of each machine tool axis. This problem can be more complex on parallel kinematic machines (PKMs) which offer multiple joint actuators. Also, it is common in industry that machines suffer from reduced operational workspace due to presence of internal singularities and self-collisions. The ability of a machine to locate its tool tip within the machine tool workspace, without collisions or singularity effects is defined as kinematic capability. Figure 3-7 represents the structure of kinematic capability designed.

(ii). Feed and spindle drives capabilities: Feed drives are used to locate the machine tool components carrying the cutting tool and part to the programmed location; therefore, their positioning accuracy determine the quality and productivity of machine tools. Feed and spindle drive capabilities represent the requirements of feed and spindle drives to cut the material out of the workpiece. Feed drive capability covers feed drive specifications such as the range of cutting speed, resolution and feed force. Spindle drive capability represents the Z-axis specifications such as power consumption and torque rate. Various types of feed drives used in CNC manufacturing has been reviewed by Altintas et al. (2011). EXPRESS-G diagrams for feed and spindle drives capabilities have been graphed in 3-8 and 3-9.



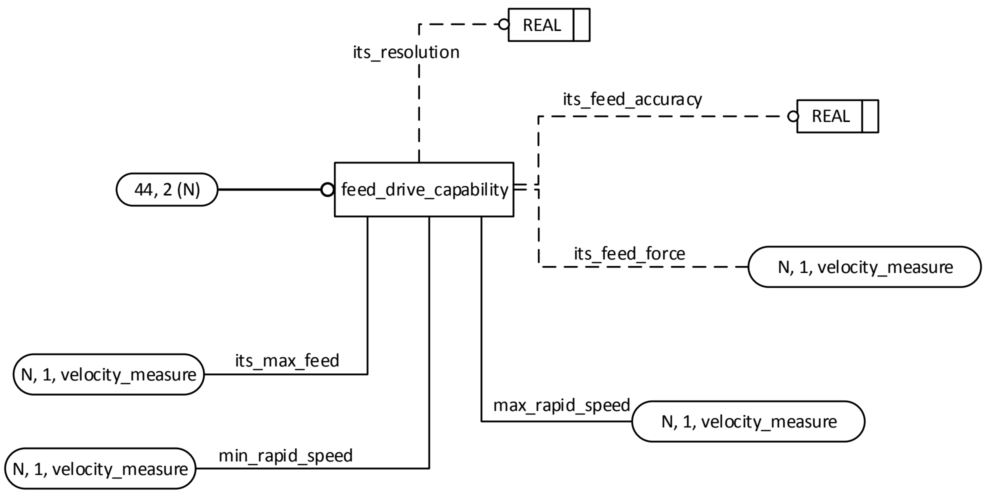
three_axis_kinematic : This attribute specifies a mechanism which represents the kinematic structure of the three axes machine tool.

four_axis_kinematic : This attribute specifies a mechanism which represents the kinematic structure of the four axes machine tool.

five_axis_kinematic : This attribute specifies a mechanism which represents the kinematic structure of the five axes machine tool.

parallel_kinematic : This attribute specifies a mechanism which represents the kinematic structure of the parallel kinematic machine.

Figure 3-7: kinematic_capability entity representation



its_resolution : This attribute specifies the smallest increment of scale which linear axes can travel.

its_max_feed : This attribute specifies the maximum programmable cutting feed.

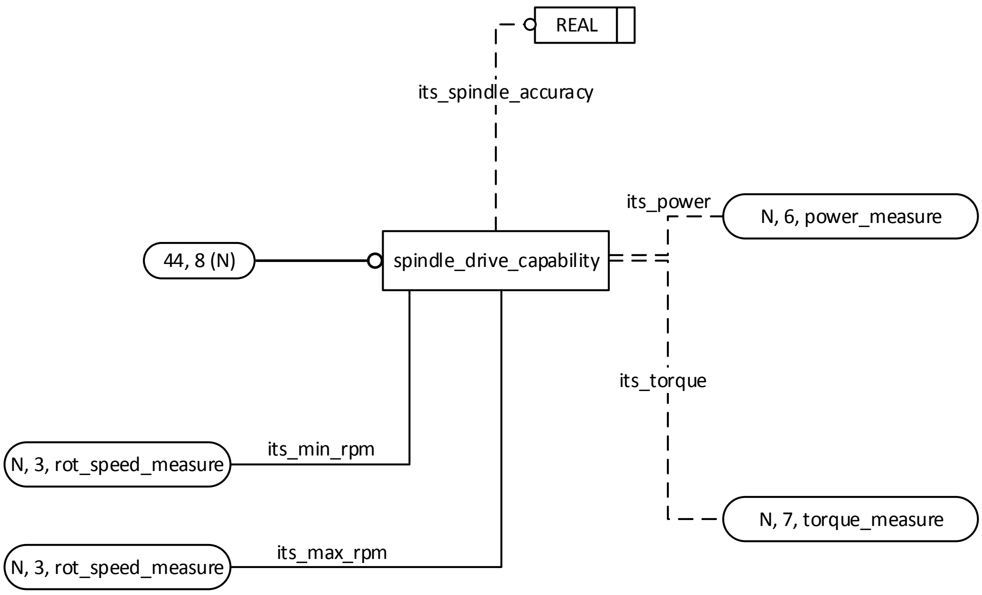
min_rapid_speed : This attribute specifies the minimum rapid traverse speed.

max_rapid_speed : This attribute specifies the maximum rapid traverse speed.

its_feed_accuracy : This attribute specifies the accuracy of the feed drive stated in the machine too handbook.

its_feed_force : This attribute specifies the cutting force generated from the feed drive.

Figure 3-8: feed_drive_capability entity representation



its_spindle_accuracy : This attribute specifies the accuracy of the spindle drive stated in the machine tool handbook.

its_min_rpm : This attribute specifies the minimum drive speed of the spindle during the operation.

its_max_rpm : This attribute specifies the maximum drive speed of the spindle during the operation.

its_torque : This attribute specifies the drive torque of the spindle during the operation.

its_power : This attribute specifies the maximum spindle power during continuous operation.

Figure 3-9: spindle_drive_capability entity representation

(iii). Tooling capabilities: This part of the machining capability profile stores the current condition of available cutting tools in a database. Cutting tools have significant effect on surface roughness of the machined part (Zahid et al., 2014). Process planners normally program machines in CAM based on the nominal values they can find on their old programs or files. However, reconditioned or worn tools do not have the nominal values on the shop-floor. In this situation, CNC operator enters the new values for cutter radius and cutter length manually or using sensors. This helps the machine to follow the compensated tool path. Having access to the actual tool dimensions can help process planners to enter actual value for tools at an earlier stage.

ISO 14649 Part 111 defined milling tool data models which can be incorporated with any STEP-NC files. With the cutter definition in ISO 14649-111, each feature can be linked with a specific tool. The properties of this tool can be saved in a separate data model in database named cutting tool dictionary in this research.

Each tool needs a tool holding method to be fixed in the machine. The large majority of tool holders clamp cutting tools, while some tool holders may attach to another tool holder for extending cutting tool length. Generally, each tool holder has a tool

Table 3.2: Machining tool selection based on workpiece material

Workpiece material	Insert material
Machining ferrous materials with the Rockwell hardness between 40 and 70	PCBN
When good surface quality is required	Cermet
General machining of aluminium, cast iron and non-ferrous metals	Uncoated carbide
Woods, plastics and composites	Tungsten carbide

locating socket and shank. A tool locating socket holds another machine element and shanks is used to locate a tool holder in another tool locating socket (Vichare et al., 2009). Also, recently introduced tooling aggregates can be used for soft materials such as solid wood, MDF, aluminum. This improve the manufacturability of these materials by adding various cutting angles on machine head.

There are some general rules in any manufacturing system which can be adapted for decision-making. For example, Table 3.2 lists appropriate tooling for cutting different materials. Also, the geometry of the machining feature requires the following attentions (Sang and Xu, 2013):

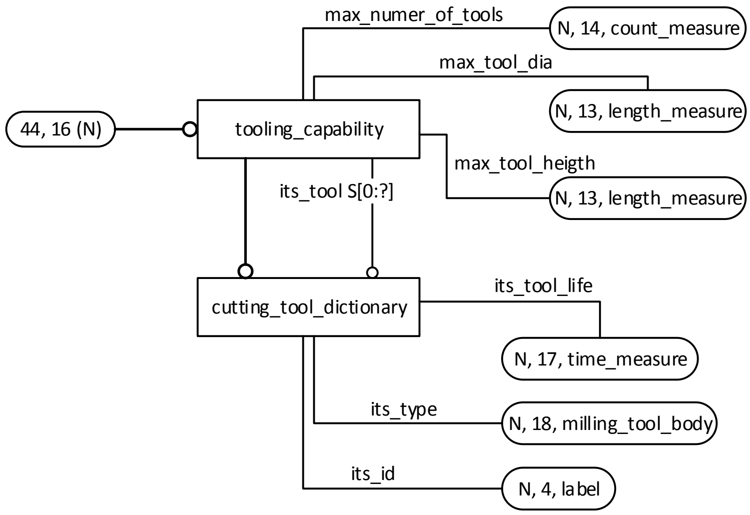
1. The radius of the cutter should be smaller than the orthogonal radius of the feature being machined.

2. The radius of ball end mill should be smaller than the planar radius of the pocket bottom

3. The ball end mill should not be used for cutting a hole with a conical bottom.

EXPRESS-G diagram for tooling capability has been graphed in Figure 3-10.

(iv). Working area capability: This part of machining capability profile represents the ability of a machine to locate a part inside its envelope. The actual positioning capability of axes has to be defined in this profile as well as real working boundary above the machine worktable. Large companies have various machine tools with different working area, and selecting the best option for machining needs to have experience about the actual machining area that machines can deliver. This can be done by understanding the accessible points within the workpiece surface by the cutter. Also, different setups of a workpiece may result in different tolerances. EXPRESS-G diagram for working area capability has been graphed in Figure 3-11.



max_number_of_tools : This attribute specifies the maximum number of tools that a tool magazine can store.

max_tool_dia : This attribute specifies the maximum diameter of tool that a tool magazine can store.

max_tool_height : This attribute specifies the maximum length of tool that a tool magazine can store.

its_tool : This attribute specifies the information needed for description of cutting tools for milling.

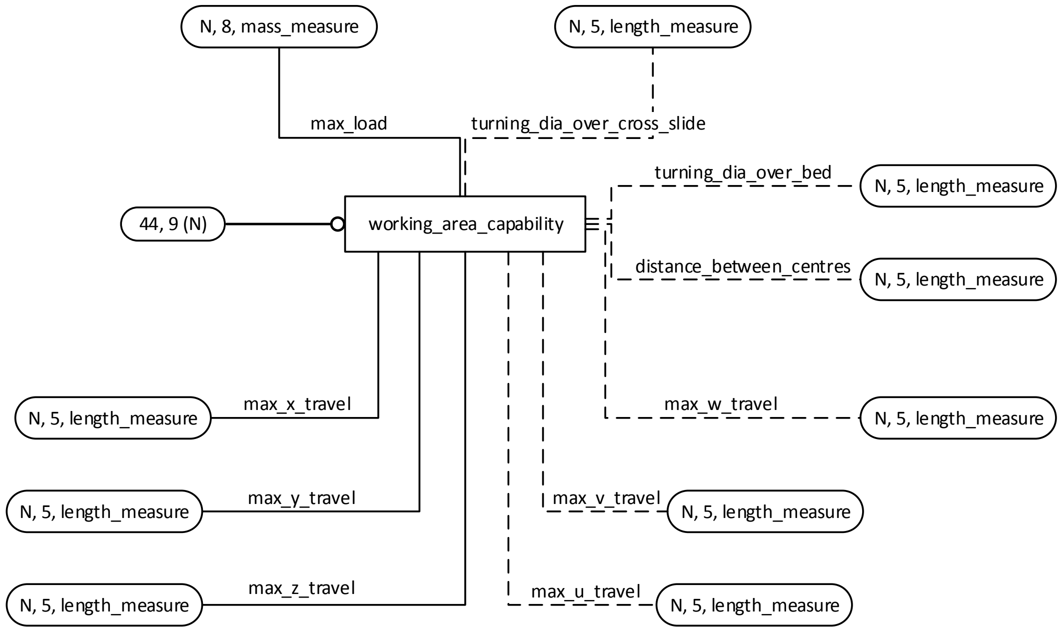
its_tool_life : This attribute specifies the nominal tool life of a milling tool.

its_type : This attribute specifies the type of the tool defined in ISO 14649-111.

its_id : This attribute specifies the tool id that identifies the tool.

Figure 3-10: tooling_capability entity representation

(v). Axes capability: is the ability of machine tool to place machine tool's axes in the accurate and repeatable location is termed axes capability. For any type of machine tools, accuracy and repeatability can be degraded during machine tool's operational life cycle. The possibility of compensating linear and rotary displacement from actual point has to be measured with periodic ball bar and laser interferometer test. These equipments can be used to periodically check the status of machine tools in shop-floor. After experiment, critical errors have to be ranked and identified in terms of their impact on machine's ability to position accurately. The ball bar test represents 18 different errors sources such as squareness, cyclic error, stick-slip, reversal spikes, scale mismatch, machine vibration, servo mismatch and backlash standardised by ISO 230-1 (2012) and ASME B5.54 (2005). Figures 3-12, 3-13, 3-14 and 3-15 represent this entity in detail.



max_load : This attribute specifies the weight of the machine.

max_x_travel, max_y_travel, max_z_travel : These attribute specify the maximum programmable axis travel of each linear axis for milling operation.

turning_dia_over_cross_slide : This attribute specifies the maximum diameter that a lathe machine can revolve over cross slide.

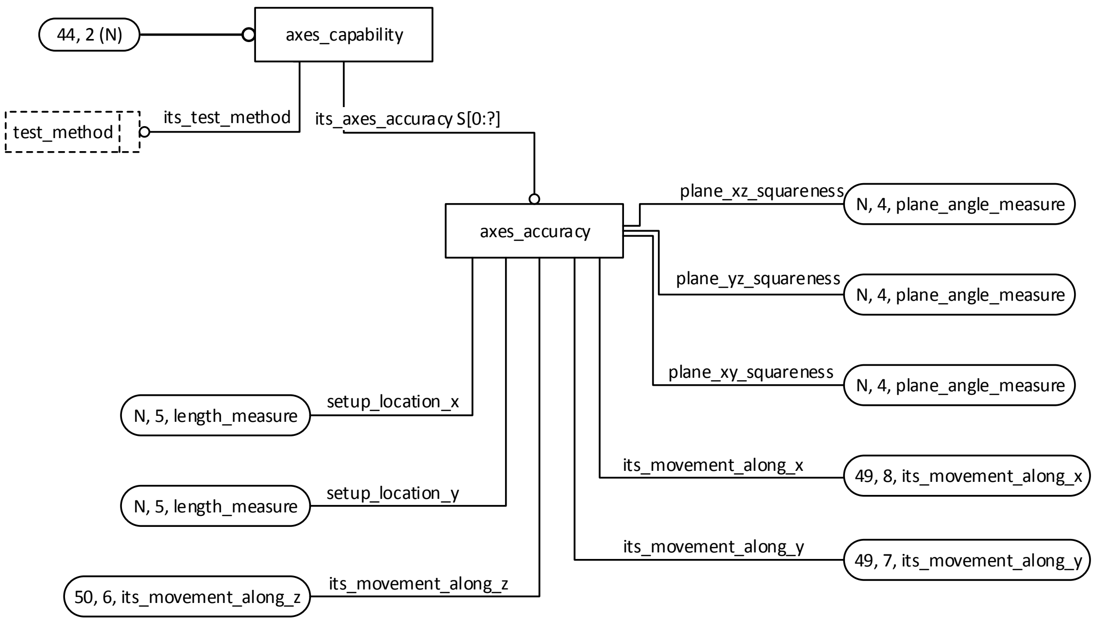
turning_dia_over_bed : This attribute specifies the maximum diameter that can be turned on a lathe without hitting the lathe bed.

distance_between_centres : This attribute specifies the distance between the centre of headstock and the centre of tailstock.

max_v_travel, max_u_travel, max_w_travel : These attribute specify the maximum programmable axis travel of each linear axis for turning operation.

Figure 3-11: working_area entity representation

(vi). Overall capability: This contains various capabilities of machine tool elements for selected machining operations. The capabilities of individual machine tool element such as controller model, its semantics and number of axes can influence on the total capability of machines. Process plans generated based on the capability profile of these elements, are more compatible with controller semantics and also machine know-how for manufacturing a part. EXPRESS-G diagram for overall capability has been graphed in Figure 3-16.



its_test_method : This attribute specifies the method used to capture the accuracy of the machine axes such as ball bar and laser interferometer.

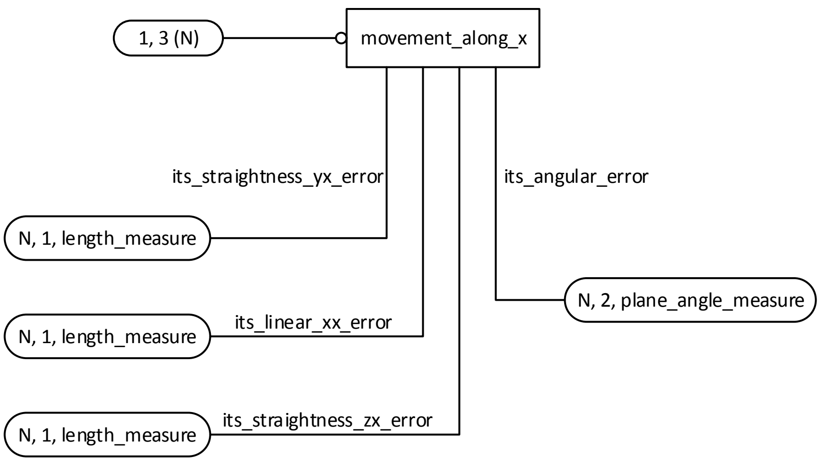
its_axes_accuracy : This attribute specifies the set of accuracy that machine axes can deliver.

plane_xz_squareness, plane_yz_squareness, plane_xy_squareness : These attributes specify the captured squareness values in XZ, YZ and XY planes.

its_movement_along_x, its_movement_along_y, its_movement_along_z : These attributes specify the captured straightness values along X, Y and Z directions.

setup_location_x, setup_location_y : These attributes specify the setup point of the test device across the machine bed.

Figure 3-12: axes_capability entity representation



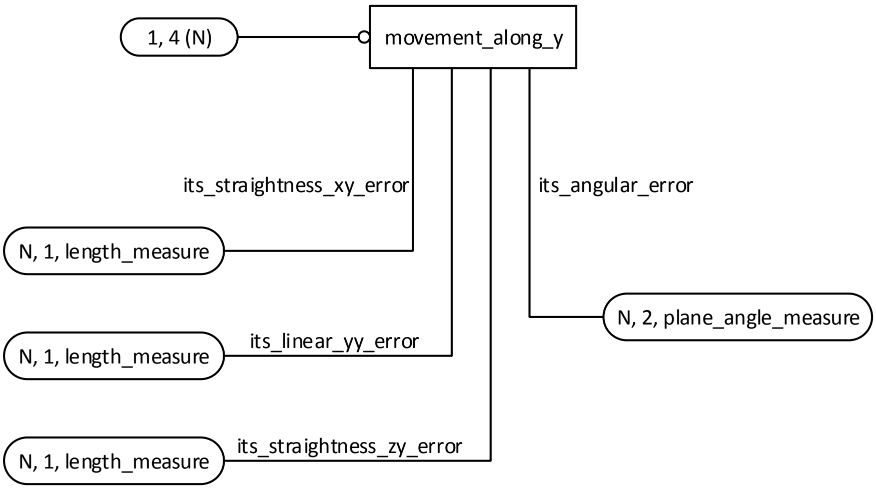
its_straightness_yx_error : This attribute specifies the straightness value of X in Y direction.

its_straightness_zx_error : This attribute specifies the straightness value of X in Z direction.

its_linear_xx_error : This attribute specifies the linear positioning error in X direction.

its_angular_error : This attribute specifies the angular error motion around A axis.

Figure 3-13: movement_along_x entity representation



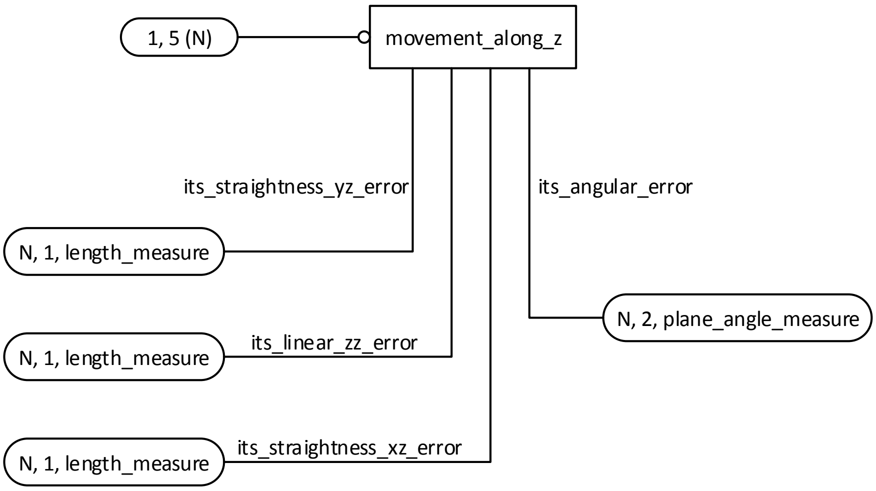
its_straightness_xy_error : This attribute specifies the straightness value of Y in X direction.

its_straightness_zy_error : This attribute specifies the straightness value of Y in Z direction.

its_linear_yy_error : This attribute specifies the linear positioning error in Y direction.

its_angular_error : This attribute specifies the angular error motion around B axis.

Figure 3-14: movement_along_y entity representation



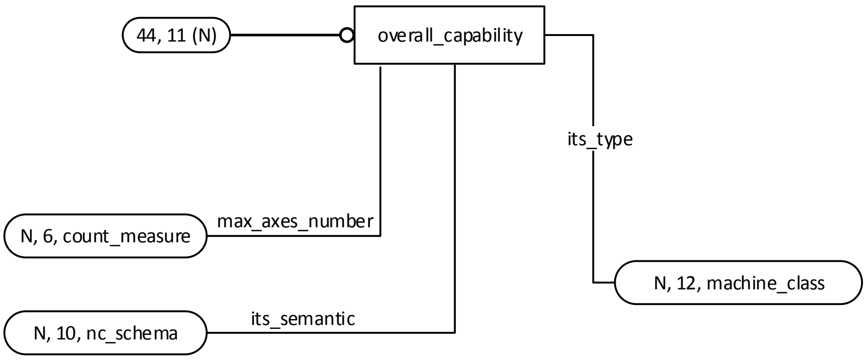
its_straightness_xz_error : This attribute specifies the straightness value of Z in X direction.

its_straightness_yz_error : This attribute specifies the straightness value of Z in Y direction.

its_linear_zz_error : This attribute specifies the linear positioning error in Z direction.

its_angular_error : This attribute specifies the angular error motion around C axis.

Figure 3-15: movement_along_z entity representation



max_axes_number : This attribute specifies the number of axes controlled simultaneously by the NC controller.

its_semantic : This attribute specifies the properties of the machine tool numerical controller.

its_type : This attribute specifies the classification of the machine tool based on its main function.

Figure 3-16: overall_capability entity representation

The UML activity diagram of the tests required for capturing the capability profile of a CNC machine has been depicted in Figure 3-17. Figure 3-17 shows the flow of information to capture the capability of a machine using CAPPable. Figure 3-17 also presents the MCP libraries developed to determine the capability of machine tool components. First, the information required to check the overall capability of a machine such as the number of axes is delivered to the next stage which checks the working envelope of the machine. After both are approved, cutting tools and cutting feeds and speeds are selected. Next, the capability of axes is assessed such as the accuracy of spindle and feed drives. Finally, the structure of the machine is analysed to determine the kinematic capability of the machine. After all of these criteria passed, CAPPable decides whether it is possible to manufacture the part or not. The decision points are placed after the each of the capability checks. Thus, CAPPable will not proceed if any of these capabilities can not be provided by the machine.
Companies with various types of machine tools can have access to their machine tool health data through the MCP libraries defined above. Additionally, informing factories with their current manufacturing capability enables them to re-use from their machining resources that have been degraded over time. Another end-user of the above framework could be the new generation of intelligent NC controllers. Intelligent NC controllers are able to adapt themselves to the recent MCP file stored in controller to respond quickly and efficiently in product life cycle domain.

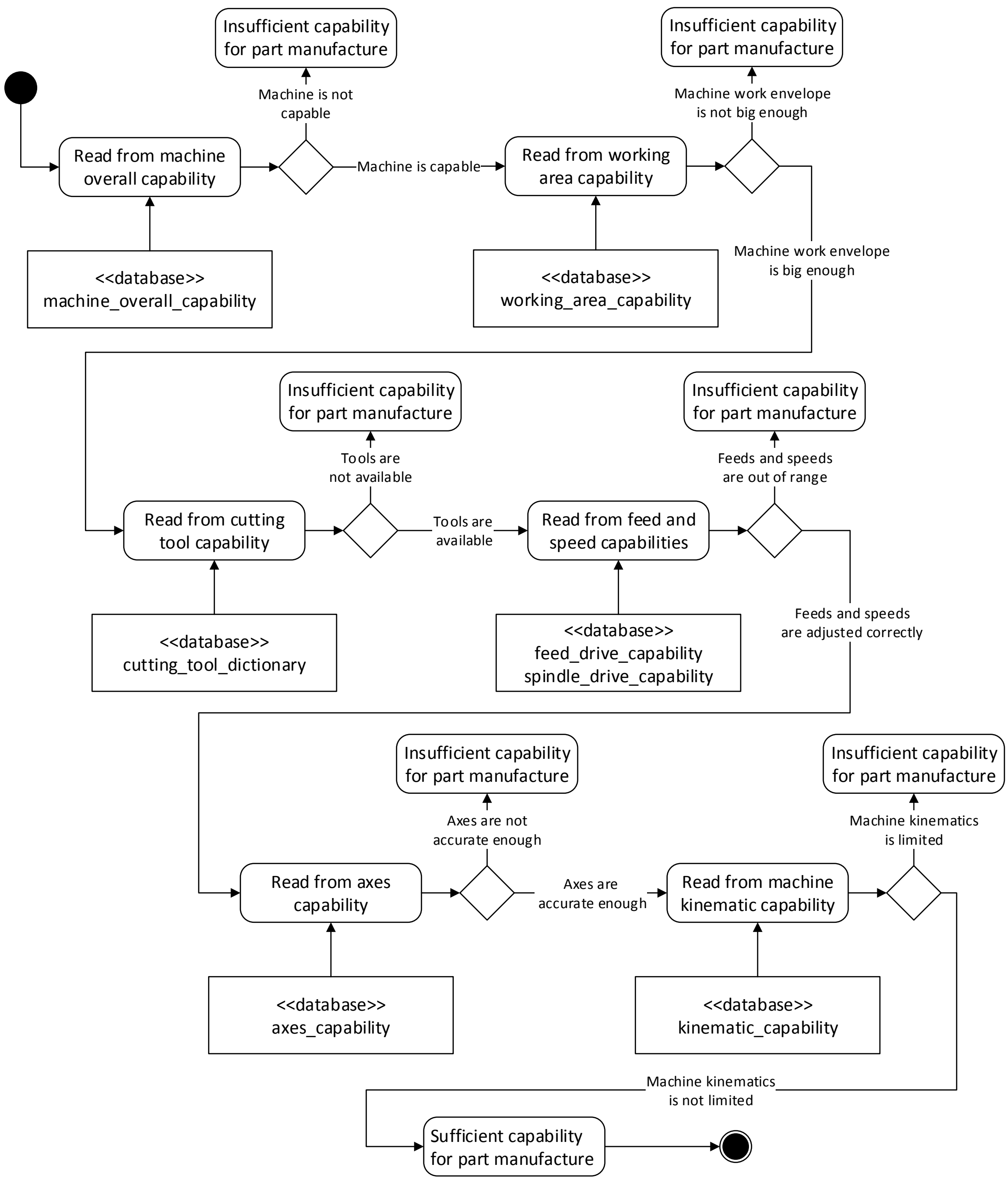


Figure 3-17: UML activity diagram of manufacturing capability checks

3.5 Computer aided process planning system

The CAPPable framework has been implemented on a Java platform, that has been developed for STEPMAN project (STEPMAN Project, 2015). This implementation contains all the data resources to implement CAD/CAM/CNC transformations. The system design of CAPPable includes two main parts, STEP-NC platform and MCP file which use EXPRESS language to define manufacturing entities. Both use object-oriented database language for ease of use in Java programming environment.

3.5.1 The EXPRESS-based MCP data model

Developed MCP schema contains following EXPRESS elements:

- Entity: is the datatype used to define data models in EXPRESS. Entities can be related to each other in a sub-supertype networks or by the specification of attributes. The following example shows the description of 'feed_drive_capability' Entity:

```
ENTITY feed_drive_capability
SUBTYPE OF (machining_capability);
min_rapid_speed : velocity_measure;
max_rapid_speed : velocity_measure;
its_feed_force : OPTIONAL velocity_measure;
its_feed_accuracy : OPTIONAL REAL;
its_resolution : OPTIONAL REAL;
its_max_feed : velocity_measure;
END_ENTITY;
```

- Defined datatypes: creates a new datatype based on another datatype. This is done by adding constraints to the previously defined datatypes and storing them with new names. For example the following example defines a new datatype which is of type 'REAL':

```
TYPE velocity_measure = REAL;
END_TYPE;
```

- Enumeration: can be used when an item of data can only get a value from a limited number of strings. In the case that an 'ENUMERATION' type is declared extensible it can be extended in other schemas. The following example shows how an 'ENUMERATION' is used to define the different types of machine tools:

```
TYPE machine_class = ENUMERATION OF (DRILLING_MACHINE,
GUNDRILL_MACHINE,
MACHINING_CENTRE,
MILLING_MACHINE,
MULTI_TASKING_MACHINE,
TURNING_MACHINE);
```

- Select: is used to select between different Entity types. This allows an attribute or variable to be one of several possible values. In the following example '`SELECT`' is used to define the possible types of values for dimensioning:

```
TYPE angle_or_length = SELECT
(plane_angle_measure,
length_measure);
END_TYPE;
```

- Simple data types: These data types are defined to represent simple containers for individual items of data such as String, Binary, Boolean, Integer and Real. For example, '`length_measure`' accepts all the real values in MCP schema.

- Aggregation data types: It is often necessary to represent data that is aggregate of other data types. EXPRESS allows various kinds of aggregation, these are: SETs, BAGs, LISTs and ARRAYs. SETs and BAGs are unordered whereas ARRAYs and LISTs are ordered. It is possible to set lower and upper limits for the size of the aggregate when defining it in EXPRESS. In the description of machine capability profile, only SETs were used. The following example shows how SET aggregate is used to define '`tooling_capability`' for a machine:
In this example, '`cutting_tool_dictionary`' is a SET of tools available for the machine.

```
ENTITY tooling_capability
SUBTYPE OF(machining_capability);
max_tool_height :  length_measure;
max_tool_dia :  length_measure;
max_number_of_tools :  count_measure;
its_tool :  SET [0:?]  OF cutting_tool_dictionary;
END_ENTITY;
```

3.6 Summary

In order to automate the manufacturing process planning system, the knowledge of the actual machining capability has to be obtained. This has been defined as one of the research gaps in Chapter 2. Thus, an EXPRESS definition of the machine tool health parameters has been presented extended by the machine tool definition. The proposed framework is designed to capture and store the actual capability of machine tools.

Chapter 4

Prototype implementation of CAPPable

4.1 Introduction

Decisions to machine a part based on the existing manufacturing resources have to be made based on the actual capability of machine tools. CNC machines normally come in many shapes and sizes with various configurations that separate them in the capabilities they can deliver. These machines vary in terms of the number of axes of movement, the arrangement of the spindles and the kinematic configuration of the moving components. Capturing the real capability of machines requires a comprehensive representation of machine tool components and cutting tools as well as part descriptions. The requirement to represent part information has been fully fulfilled with the emergence of STEP-NC standard. Currently, STEP-NC does not contain any machine tool health information.

For example, STEP-NC does not provide the engineering process planners with the available cutting tools, and the STEP-NC codes need to be adjusted by an experienced engineer. Access to the cutting tool data such as tool diameter, cutting tool length and cutting edge direction allows process planners to plan based on the available tools in machine tool bank. An extension to the STEP-NC standard has been proposed in the previous chapter to convey machining resource information to CAPPable. In this chapter, rule-based reasoning techniques have been applied on the proposed data model to assess the machineability of a part. Formal rules have been defined in Java to investigate the feasibility of the CAPPable framework. These rules are created with the aim to be a reasoning engine of the various machining resources in process planning. An object-oriented computational platform has been developed which allows planner to have access to the machine tool capability parameters. As a result, The CAPPable prototype has been generated containing sufficient machine tool health information for the part manufacturing.

4.2 Development for the implementation of the CAPPable prototype

A comprehensive MCP schema has been developed in EXPRESS language which supports ISO 14649 and AP 203 schemas. The MCP schema has been formally validated with the SDAI interface (ISO 10303-22, 1998). This formal validation searches for two common errors in EXPRESS schema: data model integrity and syntax errors such as semicolons. SDAI (ISO 10303-22, 1998) is an Application Programming Interface (API) for reading, writing and runtime manipulation of object-oriented data defined by an EXPRESS based data model. A Java based SDAI has been used to validate MCP schema, that is, files prepared according to EXPRESS standards.
Considering that EXPRESS entities in ISO 14649 are defined in an object-oriented like manner the translation does not need to substantially change the semantics involved in the creation of the models; in most cases it is only a matter of translating the data models' syntax. A prototype of the proposed framework has been generated on the Java platform. It is important to note that EXPRESS only defines states and no behaviours and therefore no methods can be derived directly from the data structures contained within an EXPRESS schema. Thus, EXPRESS entities have been translated into Java classes to implement the prototype of CAPPable.

4.2.1 Java based platform for CAPPable

This platform has been originally developed by Aydin Nassehi in Java programming language as a part of STEPMAN project (STEPMAN Project, 2015). This section provides a complete detail of the designed platform. The realisation of this platform makes it possible to run CAPPable throughout this research.

In order to translate the EXPRESS schema into a Java class collection, it has been necessary to consider the various elements of the schema as follows:

(i). Entities: The definition of entities in CAPPable begins by creating Java interfaces. The Java interface, provides prototype methods for setting the values of the class attributes and getting these values. For example the ISO 14649 entity '`two5D_manufacturing_feature`' is defined in EXPRESS as:

```
ENTITY two5D_manufacturing_feature
ABSTRACT SUPERTYPE OF (ONEOF(machining_feature, replicate_feature,
compound_feature))
SUBTYPE OF (manufacturing_feature);
feature_placement:  axis2_placement_3d;
END_ENTITY;
```

The equivalent Java interface for the above entity would be:

```
public interface two5d_manufacturing_feature_interface extends
    ExpressInterface,manufacturing_feature_interface {


 void add_its_operations(machining_operation_interface...
     _its_operations);

 void set_its_id(string_interface _its_id);
 void set_its_workpiece(workpiece_interface _its_workpiece);
 void set_its_operations(ArrayList<machining_operation_interface>
     _its_operations);
 void set_feature_placement(axis2_placement_3d_interface
     _feature_placement);

 string_interface get_its_id();
 workpiece_interface get_its_workpiece();
 ArrayList<machining_operation_interface> get_its_operations();
 axis2_placement_3d_interface get_feature_placement();
}
```

The attributes that are aggregated with LIST[?:?] and SET[?:?] in EXPRESS are both translated into ArrayLists in Java. The relationships form supertype-subtype hierarchies have been considered with Java interfaces.

(ii). Types: The general definition of simple types takes the following from in EXPRESS:

```
TYPE type_name = type_definition;
END_TYPE;
```

Where '`type_name`' is the name of the type being defined and '`type_definition`' is anther type or a primitive type. The primitive types in EXPRESS are INTEGER, STRING, BOOLEAN and REAL among others. The primitive types are defined as classes extending '`ExpressInterface`' interface that are designed for allowing multiple inheritance when necessary.

(iii). Constants: Constants can be defined as those in the '`ExpressInterface`' interface.

(iv). Functions: Functions in EXPRESS considered as methods in Java classes.

(v). Where, Derive, Inverse: WHERE, DERIVE and INVERSE are used in EXPRESS to defined rules. These rules can be implemented as validation methods in each class to ensure the integrity of the information.

(vi). Optional: in an EXPRESS schema, attributes can be defined as optional or obligatory. Java constructors have been used to allow an optional object to be created unless the non-optional attributes are set.

4.2.2 EXPRESS translator

Generating the Java classes for mappings from each EXPRESS schema is a tedious and repetitive job. It was therefore decided to employ the Software Product Lines (SPL) (Northrop and Clements, 2001) methodology to automate the process of generation of the Java mapping for each EXPRESS schema. Thus, an EXPRESS translator has been made which is capable of reading the EXPRESS schemas with the format of a text file. This application then generates sufficient Java packages to manipulate the EXPRESS files. The objects created using EXPRESS translator can then get values to run comparative tests between machines.

4.2.3 Generating MCP files

To make it easier for users to generate machine tool profiles, A STEP file generator has been used. MCP libraries have been added to this platform to read and write machine tool profiles in Java. The defined MCP libraries interfaced with STEP-NC libraries which contain '`machining_workingstep`', '`manufacturing_feature`' and '`milling_cutting_tool`'. MCP classes then interfaced with the main Java program developed in this work in order to get values and generate results. Also, getters and setters defined to obtain or add values to the entities defined in MCP libraries. In order to write an MCP file, the '`Population`' class has to be initialised. This will make an Arraylist for all the possible MCP data which can be added to the '`Population`' class. The method of generating MCP code is described by the pseudo-code listed as follows.

for *each Object in Population* **do**
  if *Object is MCP data* **then**
    ADD Object to MCP Array
  end
end

Entities only can accept the values defined in MCP library, and null values will be ignored by MCP generator. Figure folan presents the generation of the capability adjusted process plans in relation to the prototype described in this Chapter.

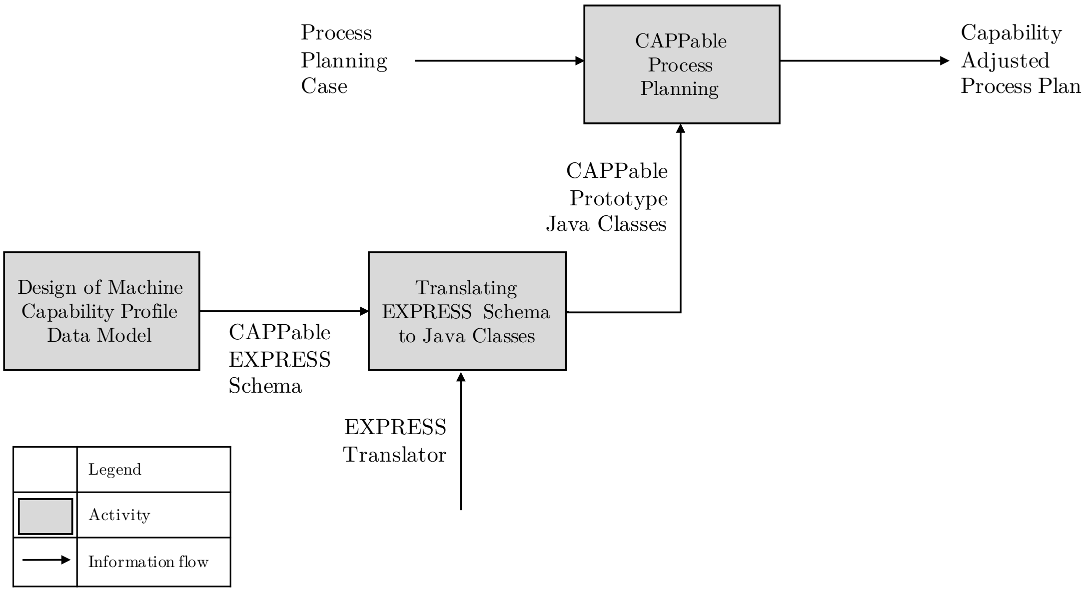


Figure 4-1: Implementation of the capability adjusted process plans using the CAPPable prototype

4.2.4 Interpreting MCP files and resource selection

Each generated '`Population`' can then be utilised to run a machine tool health check. To perform MCP interpretation, The author has developed a STEP-NC reader to read the latest MCP file released for a machine. The STEP-NC reader can use STEP-NC and MCP codes as an input in the format of text file and passes all the values to their instances. For example, the following structure is used to read an MCP file saved in '`mcp.txt`':

```
mcp.support.Population McpPop=new
   mcp.support.Population().readP21file(mcp.txt);
```

Machine tool health parameters can be ascertained from MCP files. An MCP file also contains a model of machine tool. STEP-NC codes can be read separately from a text file. The STEP-NC file transfers the design specification and CAPPable compares them with the machining requirements. Each capability parameter can be added separately to maintain the object-oriented structure of the MCP interpreter. For example, the following statements check machine working area, machine tooling and machine drives:

```
WorkingareaCapabilityCheck wa=new WorkingareaCapabilityCheck();
ToolingCapabilityCheck tc=new ToolingCapabilityCheck();
FeedDriveCapability fc=new FeedDriveCapability();
SpindleDriveCapability sd=new SpindleDriveCapability();
```

In the end, a summary of the performed checks is given to process planners in the following order:

```
Current available area in Z direction = 100.0
Required area in Z direction = 10.0
Table sizes are : [840.0, 470.0]
Workpiece geometry is : block
Required sizes to clamp the workpiece : [215.0, 265.0]
The number of available tools is : 3
The number of required tools is : 2
Available tool number 1 is a "rough mill" with the diameter of 40.0
Available tool number 2 is a "end mill" with the diameter of 40.0
Available tool number 3 is a "end mill" with the diameter of 10.0
Required tool number 1 is a "T4" with the diameter of 40.0
Required tool number 2 is a "T3" with the diameter of 10.0

Machine is capable to perform required tasks
```

The final summary provides process planners with an overview of the machining requirements which can or can not be delivered by a machine. This assessment is based on the actual condition of the machine tool.

4.3 Summary

To realise the aims of this research, a prototype of CAPPable has been presented. The CAPPable prototype has been implemented on a Java platform. The Java development and associated Java libraries have been described in detail. The generated CAPPable prototype will be deployed to run a series of experiments in the next chapter.

Chapter 5

Experiments and results

5.1 Introduction

Companies often use multiple CNCs to machine a particular type of product. CNC machines on the shop-floor constantly use for production can be degraded over time. Degraded machine tools may often lead to reduced productivity, flexibility, precision and poor responsiveness. To avoid such situations, the health condition of these physical systems must be periodically assessed. Previous chapters focused on the development of systems that provide process planners with such capabilities. This chapter explores the applicability of process planning using CAPPable through a series of experiments.

The proposed system of capturing the health of a machine has been implemented on various CNC machines in three separate sections. These series of experiments have been designed to validate the CAPPable framework. In the first section, a profile of Bridgeport and XYZ milling machines has been generated to select the most appropriate machine tool for a particular machining task. This is mainly focused on the macro process planning using CAPPable. Next section delivers the implementation of CAPPable for a PKM machine in the micro process planning stage. The third section discusses the procedure of capturing the capability of a Dugard milling machine using the ball bar test, and using the captured data for the part setup optimisation. The measured data with ball bar is used to model machining straightness and squareness errors. The best part location has been determined based on the Dugard health profile.

5.2 Investigation of capability based macro process planning

Process planning has two distinguished levels, macro- and micro-level planning (El-Maraghy, 1993). macro process planning is concerned with identifying the main tasks and their best sequence and the type of manufacturing processes, whereas micro process planning is concerned with the cutting condition determination such as adjusting feeds/speeds and calculating machining costs. The decision to machine a part based on the overall machine tool capability can be made in the macro process planning stage. The outcome of this stage can also be used for benchmarking machine tools and compare them against other CNC machines in shop-floor. Normally, not all of the available CNC machines would be able to successfully machine a part dependant on the size of the part, available cutting tools and fixtures.

Three axes machine tools are considered as the most common machines in industry. Therefore, they are used in this research to validate the proposed framework. The accuracy of a 3-axis Bridgeport machine has been checked against the machining requirements. As a result, the generated process plan ensures that the STEP-NC code has declared the actual status of machine and not nominal values. However, further expansion of the model required a comprehensive schema, which contains all the aspects of machining capability. This part of research will be explored further in Sections 5.3 and 5.4.

The health of the Bridgeport machine has been assessed to machine a rectangular closed pocket. Capability analysis of a 3-axis Bridgeport machine has been done based on the testing protocols retrieved from machine's handbook and ISO 230 series. A sample part shown in Figure 5-1 has been selected to test the proposed system. This part has been designed previously in ISO 14649 part 11 to implement STEP-NC standard.To machine the part, five working steps are required, i.e. face milling, hole drilling, hole reaming, pocket roughing and pocket finishing. The STEP-NC code generated to machine a rectangular closed pocket with its assigned tolerances and milling tools is attached in Appendix A.

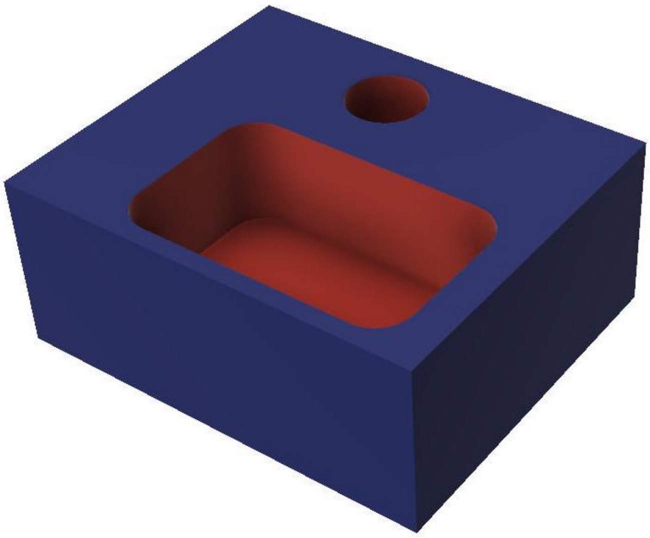


Figure 5-1: A sample part to test CAPPable framework

A MCP file has been generated based on the actual Bridgeport machine ability to deliver the required tolerances stated in the STEP-NC code. This MCP file has been attached in Appendix B. The values stored in Bridgeport capability profile approve that the machine axes have enough accuracy to finish the rectangular closed pocket. Also, the cutting tool required for this task is available in the Bridgeport tool bank. The same machining procedure has been taken to drill a hole as it has been listed in Appendix C. According to the Bridgeport capability profile, the machine is also capable of drilling the hole within the defined tolerances. 20 mm drilling tool is available in the machine tool bank to finish this job.

The capability assessment has also been done to machine the same part on a 3-axis XYZ milling machine. The associated machining capability profile has been attached to the Appendix D. The XYZ machine can not handle this job because the drilling tools are not available to finish this job. This has been concluded by comparing the STEP-NC code for the part and the generated MCP file for the XYZ machine.

5.3 Investigation of capability based micro process planning

The generation of marcoplans does not necessarily result in a globally feasible process plan for a given part (Srinivasan and Sheng, 1999). To generate reliable machining instructions, a micro process planning system that outputs the feature-level optimal plan is required. Micro process planning is a detailed process planning level in which the process planner determines the workpiece setup, machining strategies, cutting tools

and cutting conditions.
Selecting the optimised layout to setup a workpiece for a particular machining task can improve the accuracy of machining (Li and Melkote, 1999). This can be influenced by the kinematic errors of a machine tool, and the displacement errors in the machine tool axes. The hypothesis of this section is that the proposed CAPPable framework will position a workpiece on a machine bed which will result in a more accurate part than if it was randomly positioned.

The kinematic capability of a machine tool is investigated in this section. A three-axis parallel kinematic machine (PKM) which is capable of performing milling operations is used to investigate this. This machine has been selected because of its complex kinematic structure and therefore can be a good sample to validate CAPPable. A machine tool kinematic data model is introduced and linked with the existing CAPPable framework. A profile of the PKM is generated which shows the actual state of the machine. The inverse kinematic error model accounting for the axes translational error in a PKM is ascertained. The developed mathematical model is calculated and predicts the total geometric error for different positions on the machine table. Particle swarm optimisation is then utilised to determine the best location of the workpiece in terms of having the least geometric error.

5.3.1 Design of a Hybrid Manufacturing Platform

The main motivator of designing the 'Equator' platform is to have an alternative measuring system to custom gauging. The 'Equator' offers inspection of a variety of manufactured parts with different design. The main goal of adding a spindle to the 'Equator' platform is to have a hybrid platform that can perform inspection and milling operations (Kendrick, 2016). The selected platform consists of a three-axis PKM, with a dedicated controller. A detailed view of the PKM mechanism is shown in Figure 5-2. The PKM offers a smaller and lighter measuring machine compared to a traditional serial machine which requires a large cross-section to support the other axes. The PKM has a high bandwidth and in combination with the probing technology and controller design can take large quantities of scanning data per second (Kendrick, 2016).

The PKM used in this research is suspended by three drives and six constraint struts positioned about the machine workspace. These six constraint struts limit the PKM to move in rotational degrees of freedom. Also, the constraint struts are connected to moving barn doors to form parallelogram. The 'Equator' platform positions its end-effector in a three-dimensional cartesian space by extending and contracting three linear actuators. A constant attachment plate orientation is maintained through the use of three planar parallelogram structures with revolute joints. This ensures that the 'Equator' platform functions as a fully translational parallel manipulator. This type of PKM is classified as delta and the kinematic chain of it can be classified as revolute-prismatic-universal-parallelogram-universal kinematic chain (Weck and Staimer, 2002b).

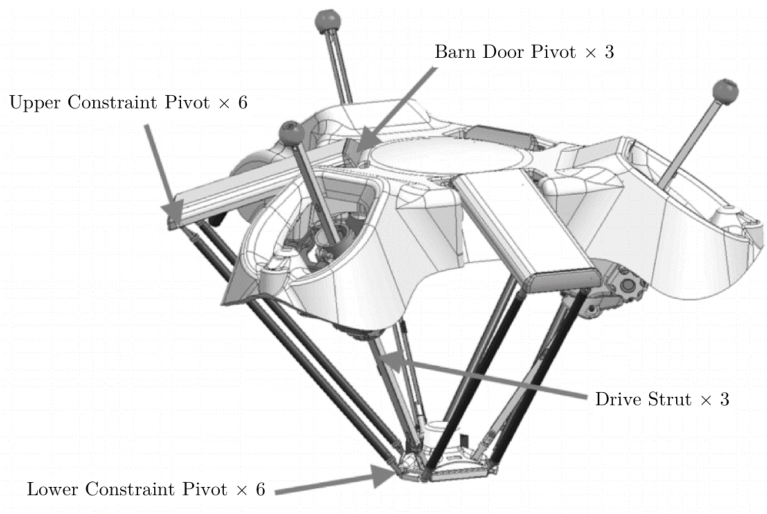


Figure 5-2: A CAD model of Renishaw's 'Equator' platform (Renishaw plc, 2018)

5.3.2 Inverse kinematics of the 'Equator'

The term 'inverse kinematics', relates to the identification of the necessary joint variable to satisfy a pre-determined end-effector position and orientation. The desired end-effector position and orientation are entered as inputs into the inverse kinematic function, which in turn outputs the necessary joint variables. The inverse kinematic function has been developed by the author.

In order to find the inverse kinematic relationship for the 'Equator' platform, a simplified model of the actuator has been considered. In this simplified model, two struts with a known variable length are connected to a common union point with $(\mathbf{x_e}, \mathbf{y_e})^\top$ coordinates; the superscript '$\top$' denotes the transposed matrix. $(\mathbf{x_e}, \mathbf{y_e})^\top$ is considered as the system's end-effector. Depending on the position of the end-effector, the extension of the linear actuators can be changed as it has been illustrated in Figure 5-3. The possible positions of the end-effector can be determined by the points of intersection between two circles. Thus, Equations 5.1 and 5.2 can be used to determine any movements of l_1 and l_2 which locate end-effector in the XY plane.
Using the same method outlined in this section, it is possible to expand this model simplification into three dimensions.

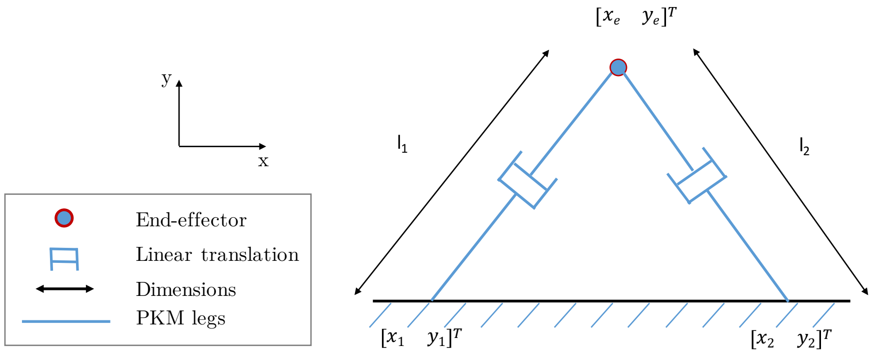


Figure 5-3: Simplified kinematic model of the Renishaw manipulator

$$(x_1 - x_e)^2 + (y_1 - y_e)^2 = l_1^2 \tag{5.1}$$

$$(x_2 - x_e)^2 + (y_2 - y_e)^2 = l_2^2 \tag{5.2}$$

5.3.3 Forward kinematics of the 'Equator'

The term 'forward kinematics', relates to the identification of a robot's end-effector position given a set of joint variables for each of the kinematic chains acting on the end-effector. The joint variables indicate the extension or rotation of each of the joints within a system for a given point in time. These variables are entered as inputs into the forward kinematic function, which in turn outputs the end-effector orientation and position with respect to the frame of reference.

Figure 5-4 shows the construction of a delta type Renishaw PKM machine including frame, joints and legs. The machine profile captured from the PKM machine contains the information about the structure of the frame, legs and joints. Assuming that the end-effector may be described as a single point of union between the three actuators, and that the orientation of the end-effector is fixed the calculation of the inverse kinematic relationship is greatly simplified. As a result of this simplification, the necessary linear actuator extensions may be described as the Euclidean distance between the actuator origins and the desired end-effector position. The position of the PKM joints has been shown in Figure 5-4(a) and are as follows:

$$x_1 = 70.71mm$$

$$x_2 = 420.71mm$$

$$y_1 = y_2 = 70.71mm$$

$$x_3 = 70.71 + 350 \times \cos 60^\circ = 245.71mm$$

$$y_3 = 70.71 + 350 \times \sin 60^\circ = 373.82mm$$

$$d_1 = 550mm, d_2 = 500mm$$
$$z_1 = z_2 = z_3 = d_3 = 600mm$$

The relationship is characterised by the following set of equations:

$$(l_1)^2 = (x_e - x_1)^2 + (y_e - y_1)^2 + (z_e - z_1)^2 \tag{5.3}$$

$$(l_2)^2 = (x_e - x_2)^2 + (y_e - y_2)^2 + (z_e - z_2)^2 \tag{5.4}$$

$$(l_3)^2 = (x_e - x_3)^2 + (y_e - y_3)^2 + (z_e - z_3)^2 \tag{5.5}$$

The simplified kinematic model presented in Figure 5-4(a) can be used to estimate the best part location on the PKM bed. It can be seen that the linear actuators do not in fact share a common union point at the end-effector. This is mainly because of the lower attachment plate which is visible in the CAD rendering in Figure 5-4(b). In this research, only the translational errors inherited from the PKM drive legs shown in Figure 5-2 have been considered, and the error caused by the movement of the constraint legs has been ignored. As a result, the lower plate is fixed all the time, and does not rotate. Thus, It has been assumed that the total kinematic error of the PKM is only affected by positioning the parallel legs.

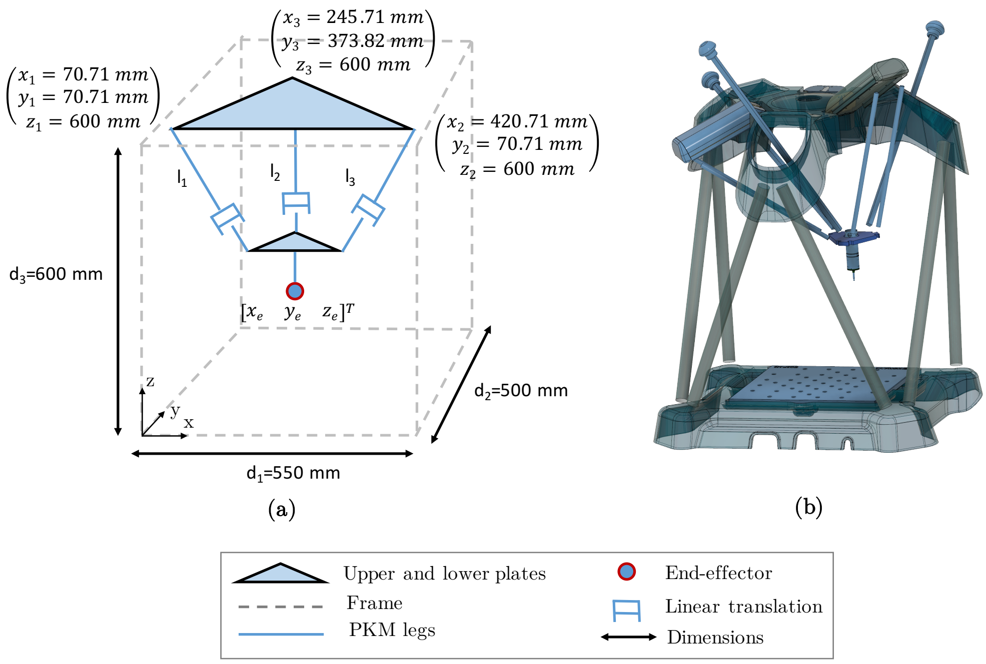


Figure 5-4: A representation of 'Equator': (a) kinematic model, (b) render

Two new vectors ($\vec{v}_1$ and $\vec{v}_2$) have been defined as it has been shown with the dashed red lines in Figure 5-5. These two vectors will transform the corner points to two new points which yield two new parallel lines. The length of the linear transformation is equal to the length of the lower plate sides (d_p). This length which is 48.5 mm for each side of the plate has been shown in Figure 5-6. Following changes will occur in Equations 5.3, 5.4 and 5.5:

$$x_2' = x_2 - 48.5mm$$
$$x_3' = x_3 - 48.5mm \times \cos 60°$$
$$y_3' = y_3 - 48.5mm \times \sin 60°$$

$$(l_1)^2 = (x_e - x_1)^2 + (y_e - y_1)^2 + (z_e - z_1)^2 \tag{5.6}$$
$$(l_2)^2 = (x_e - x_2')^2 + (y_e - y_2)^2 + (z_e - z_2)^2 \tag{5.7}$$
$$(l_3)^2 = (x_e - x_3')^2 + (y_e - y_3')^2 + (z_e - z_3)^2 \tag{5.8}$$

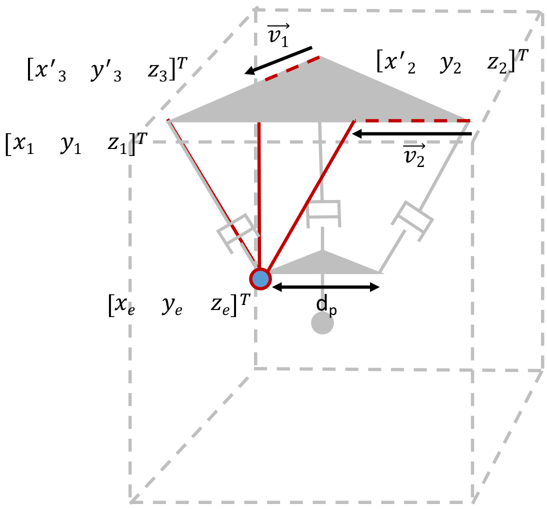


Figure 5-5: Elaborated kinematic model of the ‘Equator’ platform

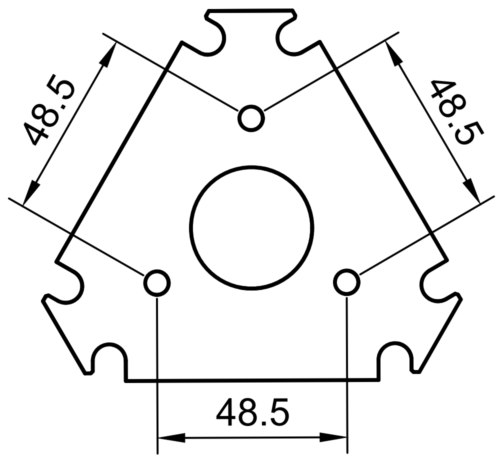


Figure 5-6: The dimensions of the ‘Equator’ attachment plate

In theory, Equations 5.3, 5.4 and 5.5 can be applied to a PKM structure to find the end-effector location. But, in practice translational errors are introduced along PKM legs which can change the position of the end-effector. Therefore, the above formulas are amended to consider the possible error factors (dl_1, dl_2 and dl_3) along PKM legs. By differentiating Equations 5.3, 5.4 and 5.5, the derivatives of the changes in the parallel legs can be modelled as follows:

$$l_1 \times dl_1 = (x_e - x_1) \times dx_e + (y_e - y_1) \times dy_e + (z_e - z_1) \times dz_e \tag{5.9}$$

$$l_2 \times dl_2 = (x_e - x_2') \times dx_e + (y_e - y_2) \times dy_e + (z_e - z_2) \times dz_e \tag{5.10}$$

$$l_3 \times dl_3 = (x_e - x_3') \times dx_e + (y_e - y_3') \times dy_e + (z_e - z_3) \times dz_e \tag{5.11}$$

Equations 5.9, 5.10 and 5.11 can be structured as follow:

$$\begin{bmatrix} l_1 & 0 & 0 \\ 0 & l_2 & 0 \\ 0 & 0 & l_3 \end{bmatrix} \times \begin{bmatrix} dl_1 \\ dl_2 \\ dl_3 \end{bmatrix} = \begin{bmatrix} x_e - x_1 & y_e - y_1 & z_e - z_1 \\ x_e - x_2' & y_e - y_2 & z_e - z_2 \\ x_e - x_3' & y_e - y_3' & z_e - z_3 \end{bmatrix} \times \begin{bmatrix} dx_e \\ dy_e \\ dz_e \end{bmatrix} \tag{5.12}$$

Finally, the location of the end-effector with respect to the linear translation of the parallel legs can be found by Equations 5.13.

$$\begin{bmatrix} dx_e \\ dy_e \\ dz_e \end{bmatrix} = \begin{bmatrix} x_e - x_1 & y_e - y_1 & z_e - z_1 \\ x_e - x_2' & y_e - y_2 & z_e - z_2 \\ x_e - x_3' & y_e - y_3' & z_e - z_3 \end{bmatrix}^{-1} \times \begin{bmatrix} l_1 & 0 & 0 \\ 0 & l_2 & 0 \\ 0 & 0 & l_3 \end{bmatrix} \times \begin{bmatrix} dl_1 \\ dl_2 \\ dl_3 \end{bmatrix} \tag{5.13}$$

By knowing the error of the coordinates in a tool path, the actual coordinates along the tool path can be calculated by Equation 5.14. x_n, y_n and z_n are the nominal tool path coordinates can be excerpted from the part program; l_1, l_2 and l_3 are the extended length of the parallel legs; x_1, x_2', x_3', y_1, y_2, y_3', z_1, z_2, z_3 can be measured from the structure of the PKM; dl_1, dl_2 and dl_3 are the translational errors along PKM legs; dx_e, dy_e and dz_e are the deviation of tool tip caused as a result of the errors in parallel legs.

$$\begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} = \begin{bmatrix} dx_e \\ dy_e \\ dz_e \end{bmatrix} + \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} \tag{5.14}$$

5.3.4 Implementation of CAPPable in micro process planning

To model a PKM actuator's capability, a parallel kinematic entity has been expressed as '`kinematic_capability`' of the machine. The data model defines PKM elements such as frame size, plate dimension and actuator legs. To incorporate the theorem established in the previous section for capturing the real capability of PKM machines, the following data models have been developed:

```
ENTITY parallel_kinematic
SUBTYPE OF(kinematic_capability);
model_name :  STRING;
parallel_legs :  SET [0:?]  OF parallel_leg;
frame_width:  length_measure;
frame_length:  length_measure;
frame_height:  length_measure;
tool_length :  length_measure;
plate_offset:  length_measure;
END_ENTITY;
```

```
ENTITY parallel_leg;
leg_id :  OPTIONAL STRING;
leg_length :  length_measure;
translational_error :  length_measure;
angle_between_adjacent_leg :  plane_angle_measure;
distance_between_adjacent_leg:  length_measure;
distance_from_frame_x:length_measure;
distance_from_frame_y:length_measure;
END_ENTITY;
```

In order to enable CAPPable to determine the optimal part location with respect to the actual available resources, it is necessary to provide resource information that reflects the status of the actuators at the time that they will be utilised for manufacturing the part. Thus, a sample capability profile has been generated for the PKM platform as follows, the full profile of the 'Equator' can be found in Appendix E:

```
#1=MACHINING_CAPABILITY_PROFILE(#10,(#2));
#2=MACHINING_CAPABILITY_DATA_POINT('"PKM PLATFORM TEST"',(#5);
#3=CALENDAR_DATE(2016,31,7);
#4=LOCAL_TIME(16,0,0.0,$);
#5=KINEMATIC_CAPABILITY(#6);
#6=PARALLEL_KINEMATIC('"RENISHAW EQUATOR PROFILE"',
(#7, #8, #9),500.0,550.0,600.0,150.0,100.0);
#7= PARALLEL_LEG('"LEG 1"', 478.0,0.02,60,350,60,60);
#8= PARALLEL_LEG('"LEG 2"',478.0,0.02,60,350,235,363);
```

```
#9= PARALLEL_LEG('"LEG 3"',478.0,0.02,60,350,410,60);
#10=MACHINE_TOOL_SPECIFICATION('"RENISHAW EQUATOR"'
,.MILLING_MACHINE.,..
```

The above profile has been used to represent the ability of a PKM machine to locate the part on its bed considering the actual capability of its actuators.

5.3.5 Artifact design for CAPPable

In order to implement CAPPable for the part setup optimisation, the sample part shown in Figure 5-7 has been selected. STEP-NC code for the part has been listed in Appendix F. The centre pocket has been selected to be machined virtually on the designed PKM platform. The accuracy of the four sides of a pocket can be a good reference for measuring the capability of a PKM machine, this is because, each parallel strut has to extend to position the tool tip within the machining area. In order to obtain X, Y and Z coordinates of the PKM end-effector, the STEP-NC code for the above model has been transferred into the G&M codes. To facilitate this transformation, the CAPPable framework has been enabled to read G&M codes in the format of plain text file.

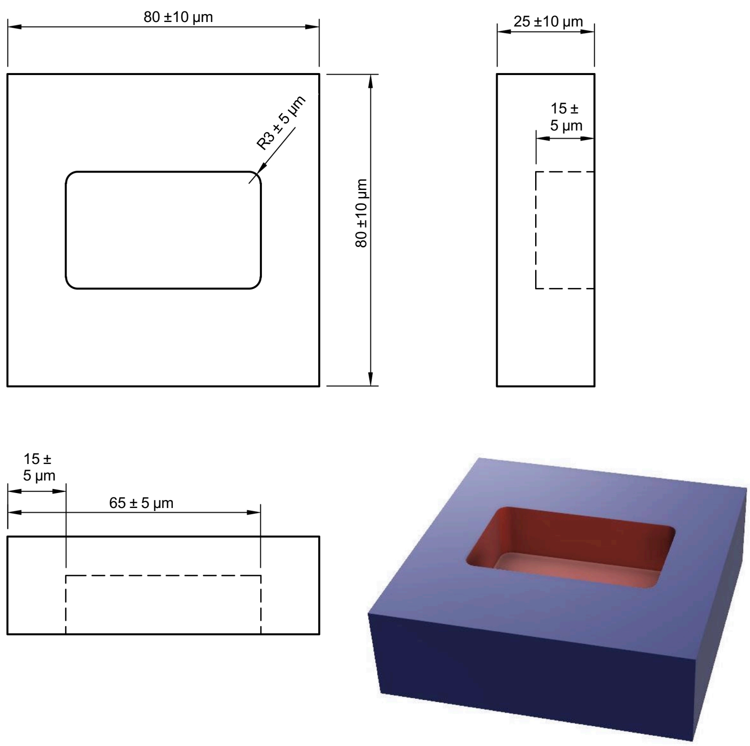


Figure 5-7: The design of a test part for PKM part location optimisation

5.3.6 Accuracy of a parallel kinematic platform

The proposed framework is tested on a parallel kinematic platform to find the best position for locating the test piece shown in Figure 5-7. In order to achieve this, it is essential to capture the current level of accuracy of the PKM. Thus, a set of errors have been assumed for each parallel leg. Considering the fact that the 'Equator' has a symmetrical design, the number of error sets can be reduced. The following errors for each PKM leg has been assumed:

$$\begin{aligned} dl_1 &= \{-20\ \mu m, -10\ \mu m, 0, 5\ \mu m, 15\ \mu m\} \\ dl_2 &= \{-4\ \mu m, 0, 12\ \mu m\} \\ dl_3 &= \{-1\ \mu m, 0, 3\ \mu m\} \end{aligned} \tag{5.15}$$

All the possibilities of having the translational errors have been considered and incorporated with CAPPable. It has been assumed that there are 45 different combinations of the leg errors for the 'Equator'. These combinations has been listed in Table 5.1. Next, using Equation 5.14, the actual position of the tool tip has been estimated accounting for the translational errors in the PKM. A PSO is used to identify the part location on a PKM machine bed that will result in the least amount of translational error in the PKM legs. Each particle in the PSO is constructed a coordinate/point to represent the static part location. The PSO is initialised with a set of points at random locations. Then, the total error for each random part location has been determined using Equation 5.16.

$$\begin{aligned} Er_i &= \sqrt{(X_{i,a} - X_{i,n})^2 + (Y_{i,a} - Y_{i,n})^2 + (Z_{i,a} - Z_{i,n})^2} \\ & \qquad i = 1, ..., 45. \end{aligned} \tag{5.16}$$

Er_i = The total error factor
$X_{i,a}, Y_{i,a}, Z_{i,a}$ = Calculated actual points
$X_{i,n}, Y_{i,n}, Z_{i,n}$ = Nominal points
The procedure outlined above has been executed over all iterations to find the optimal answer for each set of errors in Table 5.1. The algorithm of the software developed for this part of the research is as follows:

1-Input Read G-code of part in text format
Find the nominal tool tip positions from the text file
2- Generate sets of potential error values
3- Calculate the actual tool tip position for each leg
4- Calculate the difference between actual values from step 3 and the nominal tool tip positions from the G-code
5- Calculate the total error value by adding all the error values from step 4
6- Find the minimum total error value with PSO
7- Output Optimal part location

5.3.7 Optimising part location on PKM bed

To find the best location of the workpiece on PKM bed, the actual position of the tool tip has been ascertained based on the translational error along PKM legs. Then, the error factor has been determined for each point in tool path. Finally, particle swarm optimisation has been applied to try the same tool path in different locations on the PKM bed. These series of trials conducted through the generation of particles to find the least total error. The objective function has been defined as follow:

$$F_i = \sum_{j=0}^{j=n} \sqrt{(P_{i,x} + T_{j,x} - A_{j,x})^2 + (P_{i,y} + T_{j,y} - A_{j,y})^2} \tag{5.17}$$

j = Each point in the tool path
$T_{j,x}$, $T_{j,y}$ = Position of tool at tool path point j
n = number of points in tool path
$A_{j,x}$, $A_{j,y}$ = Actual position of tool at point j
$P_{i,x}$, $P_{i,y}$ = Random particles generated by PSO

The PSO constraints can be calculated by the available workspace of the 'Equator' . The accessible working volume of the PKM is mainly determined by the length of the three parallel legs. This naturally forms a three-dimensional teardrop shape. However, to make it easier for PSO to determine the optimal workpiece location, a working volume of a 300 mm (diameter) by 150 mm cylinder has been considered. Thus, the following constraints are considered for optimising the Equation 5.17:
For $P_{i,x}$ $\quad x_{min} + T_{xmin} \leq \text{x} \leq x_{max} + T_{xmax}$
For $P_{i,y}$ $\quad y_{min} + T_{ymin} \leq \text{y} \leq y_{max} + T_{ymax}$
Each optimisation runs for 10000 iterations to ensure that the optimisation converged to a solution. The particle has been found using the following PSO parameters:
Inertia weight (W) = 1
Acceleration coefficients (C_1 , C_2) =2
The Java codes for the above optimisation problem has been presented in Appendix G. The number of particles used was 1000 to ensure a good distribution of starting positions in the search space. For the test case used, the solution was found in less than 20 seconds. The Z-axis has not been included in the development of software as the main purpose is to find the optimal part location in 2D. This also helps with the computational time. The iterations for the first combination of errors is shown in Table 5.2. Table 5.3 shows the test result for all the different combinations of the axial errors listed in Table 5.1. The optimal part setup position found in $X = 0$ and $Y = 0$ which locates in the centre of the PKM bed.

Table 5.1: Various axial errors in parallel legs

Test	dl_1 (μm)	dl_2 (μm)	dl_3 (μm)
1	-20	-4	-1
2	-20	-4	0
3	-20	-4	3
4	-20	0	-1
5	-20	0	0
6	-20	0	3
7	-20	12	-1
8	-20	12	0
9	-20	12	3
10	-10	-4	-1
11	-10	-4	0
12	-10	-4	3
13	-10	0	-1
14	-10	0	0
15	-10	0	3
16	-10	12	-1
17	-10	12	0
18	-10	12	3
19	0	-4	-1
20	0	-4	0
21	0	-4	3
22	0	0	-1
23	0	0	0
24	0	0	3
25	0	12	-1
26	0	12	0
27	0	12	3
28	5	-4	-1
29	5	-4	0
30	5	-4	3
31	5	0	-1
32	5	0	0
33	5	0	3
34	5	12	-1
35	5	12	0
36	5	12	3
37	15	-4	-1
38	15	-4	0
39	15	-4	3
40	15	0	-1
41	15	0	0
42	15	0	3
43	15	12	-1
44	15	12	0
45	15	12	3

Table 5.2: PSO iterations (units are in µm)

Iteration 0	Fitness: 28825.26	X: 2619.95	Y: -2467.75
Iteration 0	Fitness: 13226.08	X: -913.05	Y: -1425.58
Iteration 0	Fitness: 8774.4	X: 221.95	Y: -1094.59
Iteration 0	Fitness: 8684.73	X: -988.36	Y: -539.00
Iteration 2	Fitness: 6851.71	X: 527.24	Y: -676.94
Iteration 4	Fitness: 4725.59	X: 550.97	Y: -128.98
Iteration 11	Fitness: 4530.49	X: 482.83	Y: -265.60
Iteration 11	Fitness: 3262.12	X: -437.79	Y: -0.34
Iteration 18	Fitness: 2355.36	X: 243.83	Y: -133.14
Iteration 31	Fitness: 1383.28	X: -52.5	Y: 144.23
Iteration 36	Fitness: 196.0	X: -19.8	Y: -5.50
Iteration 51	Fitness: 101.23	X: -18.94	Y: -31.42
Iteration 57	Fitness: 70.57	X: -25.99	Y: -34.65
Iteration 59	Fitness: 67.65	X: -28.55	Y: -19.21
Iteration 61	Fitness: 50.03	X: -28.51	Y: -21.56
Iteration 62	Fitness: 39.44	X: -35.74	Y: -26.33
Iteration 62	Fitness: 29.16	X: -33.01	Y: -24.35
Iteration 64	Fitness: 14.73	X: -29.1	Y: -27.64
Iteration 64	Fitness: 9.45	X: -29.74	Y: -27.43
Iteration 69	Fitness: 4.4	X: -31.47	Y: -27.33
Iteration 71	Fitness: 1.04	X: -30.92	Y: -27.46
Iteration 76	Fitness: 0.76	X: -30.82	Y: -27.32
Iteration 76	Fitness: 0.53	X: -30.98	Y: -27.35
Iteration 79	Fitness: 0.44	X: -30.91	Y: -27.39
Iteration 80	Fitness: 0.26	X: -30.9	Y: -27.31
Iteration 85	Fitness: 0.24	X: -30.94	Y: -27.35
Iteration 86	Fitness: 0.2	X: -30.93	Y: -27.33
Iteration 88	Fitness: 0.19	X: -30.92	Y: -27.35
Iteration 88	Fitness: 0.19	X: -30.93	Y: -27.33
Iteration 89	Fitness: 0.18	X: -30.91	Y: -27.33
Iteration 89	Fitness: 0.18	X: -30.92	Y: -27.33
Iteration 90	Fitness: 0.18	X: -30.92	Y: -27.33
Iteration 92	Fitness: 0.18	X: -30.92	Y: -27.33
Iteration 96	Fitness: 0.18	X: -30.92	Y: -27.33
Iteration 99	Fitness: 0.18	X: -30.92	Y: -27.33

Table 5.3: PSO test results

Test	Fitness (µm)	Part location in X (µm)	Part location in Y (µm)
1	0.18	-31	-27.33
2	0.19	-30.1	-28.71
3	0.24	-30.91	-32.84
4	0.2	-40.32	-21.9
5	0.2	-40.34	-23.29
6	0.24	-40.33	-27.41
7	0.28	-68.54	-5.61
8	0.28	-68.54	-6.99
9	0.3	-68.54	-11.12
10	0.03	7.39	-5.22
11	0.04	7.39	-6.59
12	0.09	7.39	2.13
13	0.02	-2.02	0.21
14	0.01	-2.02	-1.16
15	0.07	-2.02	-5.3
16	0.13	-30.23	16.5
17	0.13	-30.23	15.13
18	0.13	-30.23	11
19	0.04	9.41	-4.05
20	0.04	9.41	-5.43
21	0.09	9.41	-9.56
22	0.02	0.0	1.38
23	0.01	0.0	0.0
24	0.06	0.0	-4.13
25	0.13	-28.2	17.67
26	0.12	-28.2	16.29
27	0.12	-28.22	12.16
28	0.08	19.49	1.77
29	0.08	19.49	0.39
30	0.09	19.49	-3.74
31	0.06	10.08	7.2
32	0.05	10.08	5.82
33	0.06	10.08	1.69
34	0.13	-18.14	23.49
35	0.12	-18.14	22.11
36	0.09	-18.14	17.98
37	0.18	39.65	13.41
38	0.18	39.65	12.03
39	0.16	39.65	7.9
40	0.18	39.65	13.41
41	0.15	30.24	17.46
42	0.13	30.24	13.33
43	0.18	2.03	35.13
44	0.16	2.03	33.75
45	0.12	2.03	29.62

5.4 Part setup optimisation based on machine health

In this section, the proposed framework for capturing the health of a machine has been implemented on a Dugard CNC machine. This type of CNC machine has been considered as the most commonly used machine in industry. The ball bar test based on ISO 230-1 (2012) is used to obtain the latest health condition of a Dugard machine tool in machine tool lab, and machine capability profile for part setup optimisation.
To capture the health of Dugard CNC machine for part setup optimisation, five main steps have been taken:

1. Kinematic modelling of Dugard machine tool to determine static errors effecting the machining accuracy in each tool tip position.

2. A series of ball bar tests to position across the machine table to record machine tool errors such as squareness and straightness.

3. Curve fitting of measured error values to replicate the recorded data from ball bar.

4. Identification of best part position considering straightness and squareness errors.

5. Validating error models through the machining of six test parts on the machining centre.

Following this section, each of the above steps will be explored in detail.

5.4.1 Kinematic modelling of machine tools

The actual motion of each of the machines axes consists of three pure translations and three pure rotations, corresponding to the six possible degrees of freedom of a moving body. Any points in machine coordinate system can be presented by vectors. Each of these points represents the tool tip position at an arbitrary position within the working volume of the machine. A tool tip position (P) can be represented as a single vector $\overrightarrow{OP}$ (Postlethwaite, 1992):

$$\overrightarrow{OP} = \begin{pmatrix} X_{OP} \\ Y_{OP} \\ Z_{OP} \end{pmatrix} \tag{5.18}$$

X_{OP}, Y_{OP} and Z_{OP} are actual coordinates of the point P. Alternatively, the vector $\overrightarrow{OP}$ can be expressed as a sum of a chain of three vectors. Each of these vectors are parallel to one of the axes of machine resulting in three individual vectors in X,Y and

Z directions:

$$\overrightarrow{OP} = \overrightarrow{X_{OP}} + \overrightarrow{Y_{OP}} + \overrightarrow{Z_{OP}} \tag{5.19}$$

These vectors express the movement of tool tip to reach position P considering the translational errors in each direction:

$$\overrightarrow{X_{OP}} = \begin{pmatrix} x + E_{XX} \\ E_{YX} \\ E_{ZX} \end{pmatrix} \tag{5.20}$$

$$\overrightarrow{Y_{OP}} = \begin{pmatrix} E_{XY} \\ y + E_{YY} \\ E_{ZY} \end{pmatrix} \tag{5.21}$$

$$\overrightarrow{Z_{OP}} = \begin{pmatrix} E_{XZ} \\ E_{YZ} \\ z + E_{ZZ} \end{pmatrix} \tag{5.22}$$

where x,y and z are the nominal positions of machine tool axes. E_{XX}, E_{YY} and E_{ZZ} are linear positioning errors (µm). E_{YX}, E_{ZX}, E_{XY}, E_{ZY}, E_{XZ} and E_{YZ} are straightness errors (µm).

Depending on the kinematic chain of machines, rotation of one axis can affect on other axes which link to it. Especially, for the axis which mounted upon the axis of rotation. Angular error motions around A-axis, B-axis and C-axis have been approximated and the transformation matrices ($R(x)$ and $R(y)$) modelled as follows:

$$R(x) = \begin{pmatrix} 1 & -E_{CX} & +E_{BX} \\ +E_{CX} & 1 & -E_{AX} \\ -E_{BX} & +E_{AX} & 1 \end{pmatrix} \tag{5.23}$$

$$R(y) = \begin{pmatrix} 1 & -E_{CY} & +E_{BY} \\ +E_{CY} & 1 & -E_{AY} \\ -E_{BY} & +E_{AY} & 1 \end{pmatrix} \tag{5.24}$$

where:
E_{AX} and E_{AY} are angular error motions around A-axis (µrad).
E_{BX} and E_{BY} are angular error motions around B-axis (µrad).
E_{CX} and E_{CY} are angular error motions around C-axis (µrad).
In the above calculations it was assumed that the axes of machine are perfectly square to each other. In practice there will be errors in orthogonality between mutually perpendicular axes. The orthogonality errors associated with the machine can be estimated

with the following transformation matrix:

$$R(s) = \begin{pmatrix} 1 & -E_{C0X} & +E_{B0X} \\ 0 & 1 & -E_{A0Y} \\ 0 & 0 & 1 \end{pmatrix} \tag{5.25}$$

where:
E_{C0X} is squareness error of X to Y (µm/m)
E_{B0X} is squareness error of X to Z (µm/m)
E_{A0Y} is squareness error of Y to Z (µm/m)

Different configuration of machine tools may use different combination of transformation matrices to find the actual position of tool tip. The Dugard machine tool consists of a separate Z-axis. The X and Y axes are attached to each other in kinematic chain. For this type of machine tool shown in Figure 5-8 the following matrix transformations need to be done:

$$\overrightarrow{OP} = R(x)^{-1} \times R(y)^{-1} \times R(s) \times (\overrightarrow{Y_{OP}} + \overrightarrow{X_{OP}}) + R(s) \times \overrightarrow{Z_{OP}} \tag{5.26}$$

The following considerations have been taken to structure the above expression:

1. Rotation of X-axis effects on Y-axis.
2. Rotation of X-axis effects on the datum point.
3. Rotation of Z-axis does not effect on any other axes.
4. Orthogonality errors ($R(s)$) associated with the machine have been considered in all directions.

After simplifying the expression above in MATLAB, the following error values can be calculated for each point in machine tool workspace:

$$e_x = E_{XY} - Y \times E_{C0X} \tag{5.27}$$

$$e_y = E_{YX} \tag{5.28}$$

where Y is the nominal positions of y axis; E_{YX} is straightness of x in y direction and E_{XY} is straightness of y in x direction; E_{C0X} is squareness of x with respect to y in the xy plane. Second order terms have been ignored in calculating e_x and e_y because these values are small and do not change the final results.

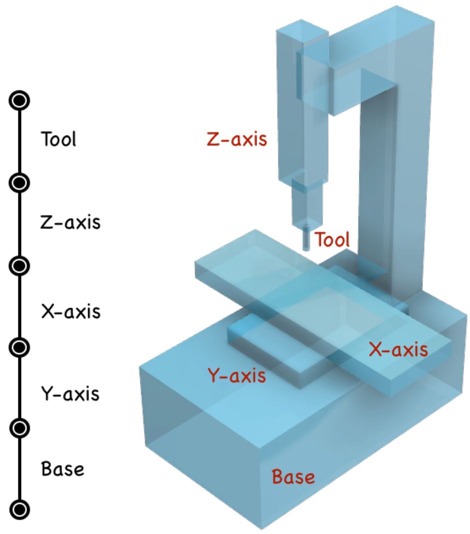


Figure 5-8: Kinematic chain of Dugard Eagle 850

5.4.2 Modelling of machining errors using ball bar

The ball bar test has been used to capture the degree of deviation the two axes of the Dugard machine tool while moving on a circular path. Various types of errors can be obtained using the ball bar test, such as backlash, straightness, squareness, servo mismatch, and reversal spikes. The results generated by the ball bar test provide a diagnostic review of the machine's capability in critical areas (Jalaludin et al., 2017). Straightness and squareness errors have been assumed as the major sources of machining error (Stephenson and Agapiou, 2016). For other machining errors such as backlash and servo mismatch, the reader is directed to research undertaken by Lei et al. (2009) and Ramesh et al. (2000a,c). These studies provide a comprehensive review of theses types of errors.

In modelling straightness, it has been assumed that the area under the test, has a simple bend or curve giving rise to a straightness error. For example, if a ball bar with 100 mm length measures a straightness error of 1.3 microns in the X direction, the straightness of x in the y direction can be estimated by $E_{YX} = 0.000000013 \times x^2$. This is illustrated in Figure 5-9.

A series of ball bar tests were required to cover the full length of a typical machine tool bed. Thus, several parabolas represent the overall straightness error. The measured data points from ball bar were fitted to Fourier models using MATLAB curve fitting toolbox to show the range of straightness errors. A combination of straightness error parabolas has been shown in Figure 5-10. Also, best fitted curve has been highlighted in red.

Squareness error has been considered as a least square line fitted for the variation in straightness deviation recorded for series of ball bar tests. Figure 5-11 shows the straightness error which has been superposed onto squareness line (red line).

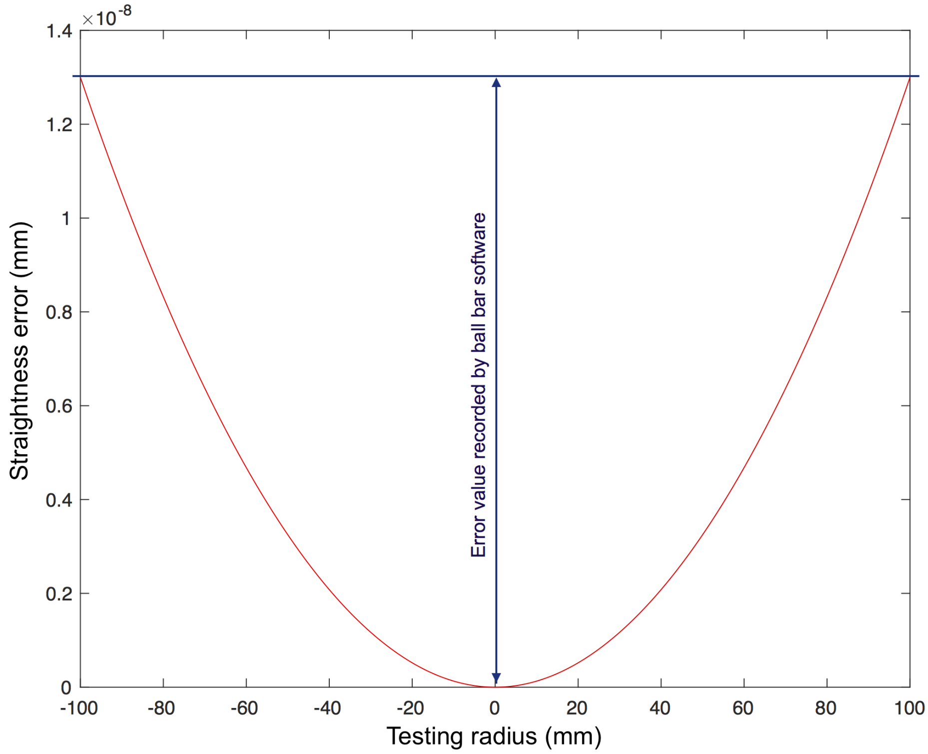


Figure 5-9: Straightness of x in y direction

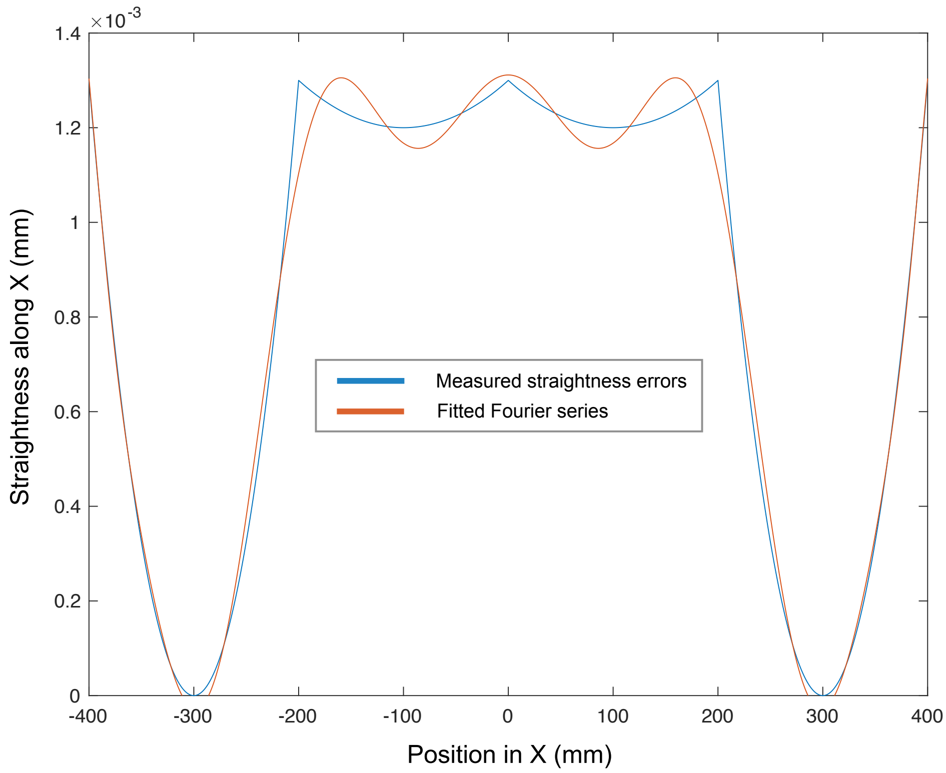


Figure 5-10: Variations of straightness along x

To test the theory mentioned in Sections 5.4.1 and 5.4.2, Renishaw QC20-W ball bar device was used in eight different locations (4×2). The ball bar test ran on a Dugard Eagle 850 machine bed to record data for modelling straightness and squareness errors. The procedure shown in Figure 5-12 has been done eight times to cover a travel range of 800mm × 400mm. The number of tests (i.e. 8) has been determined by the size of ball bar (i.e. 100 mm) and machine bed. The order of each test has been shown in Figure 5-13.

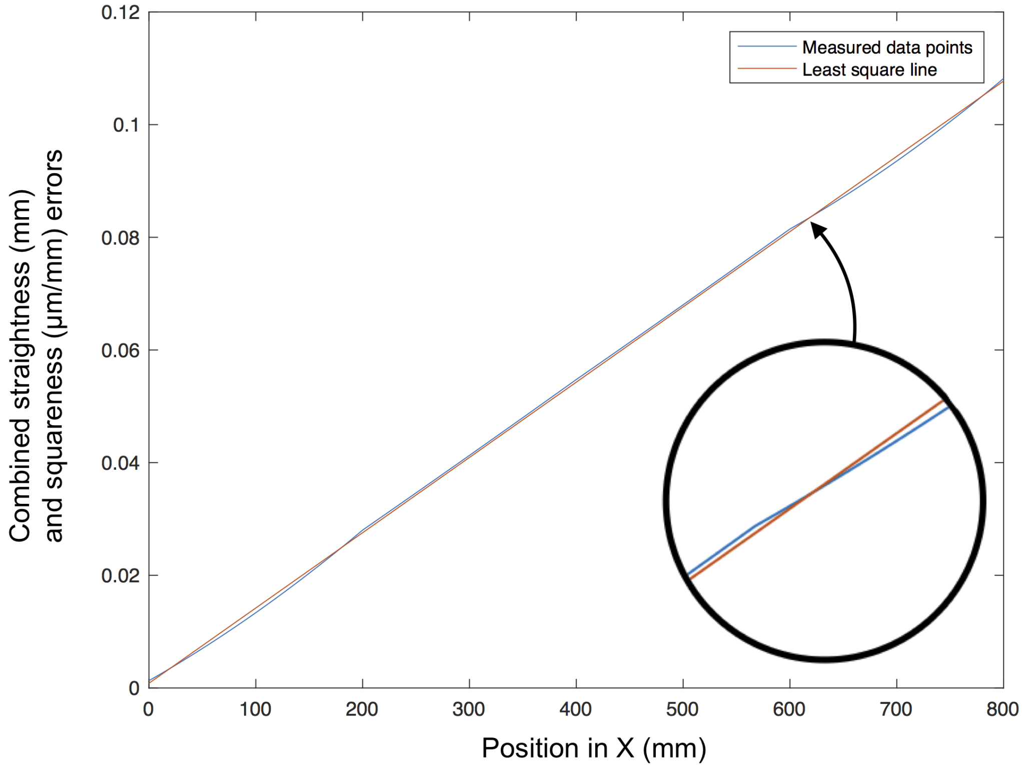


Figure 5-11: Squareness of x with respect to y in xy plane

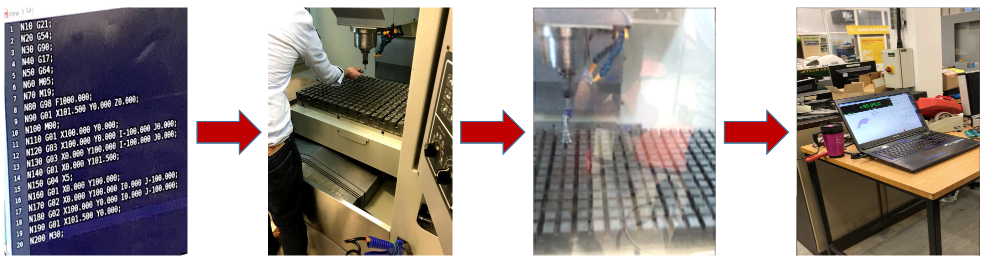


Figure 5-12: Ball bar setup and procedure

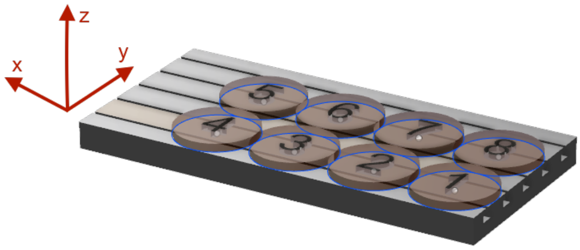


Figure 5-13: Sequence of the ball bar test on Dugard machine tool

The measured values from the above tests were used to find straightness error function for each zone. Next, Fourier series were used to formulate the straightness error function along X, and Y. Table 5.4 summarises the measured error values in 8 different locations. Tables 5.5 and 5.6 list straightness and squareness models for tests 1 to 4 and 5 to 8.

Table 5.4: The ball bar test results

Machining errors	1	2	3	4	5	6	7	8
Squareness (μm/m)	125.6	135.4	124.1	137.8	142.1	129.8	142.2	131.8
Straightness x (μm)	1.3	0.1	0.1	1.3	0.6	-0.8	0.8	0.4
Straightness y (μm)	-0.4	1.6	1.7	0.9	2	2.5	1.8	1.5

Table 5.5: Error models of straightness and squareness

Error models	1	2	3	4
E_{YX}	$1.3^{-8} \times x^2$	$0.1^{-8} \times x^2$	$0.1^{-8} \times x^2$	$1.3^{-8} \times x^2$
E_{XY}	$-0.4^{-8} \times y^2$	$1.6^{-8} \times y^2$	$1.7^{-8} \times y^2$	$0.9^{-8} \times y^2$
Fitted curve E_{YX}	-8.355 × 10^7 + 1.323 × 10^8 × cos(0.001122 × x) + 6.372 × 10^7 × sin(0.001122 × x) - 6.186 × 10^7 × cos(0.002244 × x) -7.757 × 10^7 × sin(0.002244 × x) + 1.129 × 10^7 × cos(0.003366 × x) 4.945 × 10^7 × sin(0.003366 × x) + 4.234 × 10^6 × cos(0.004488 × x) -1.855 × 10^7 × sin(0.004488 × x) - 3.088 × 10^6 × cos(0.00561 × x) 3.872 × 10^6 × sin(0.00561 × x) + 7.211 × 10^5 × cos(0.006732 × x) -3.473 × 10^5 × sin(0.006732 × x) - 6.062 × 10^4 × cos(0.007854 × x) 0.7058 × sin(0.007854 × x)			
Fitted line E_{C0X}	1.3363e× 10^{-4} × x + 8.3393 × 10^{-4}			

Table 5.6: Error models of straightness and squareness

Error models	5	6	7	8
E_{YX} E_{XY}	$0.6^{-8} \times x^2$ $0.2^{-7} \times y^2$	$-0.8^{-8} \times x^2$ $2.5^{-8} \times y^2$	$0.8^{-8} \times x^2$ $1.8^{-8} \times y^2$	$0.4^{-8} \times x^2$ $1.5^{-8} \times y^2$
Fitted curve E_{YX}	-3.971 × 10^7 + 6.252 × 10^7 × cos(0.001122 × x) + 3.106 × 10^7 × sin(0.001122 × x) - 2.851 × 10^7 × cos(0.002244 × x) -3.76 × 10^7 × sin(0.002244 × x) + 4.507 × 10^6 × cos(0.003366 × x) 2.372 × 10^7 × sin(0.003366 × x) + 2.443 × 10^6 × cos(0.004488 × x) -8.73 × 10^6 × sin(0.004488 × x) - 1.581 × 10^6 × cos(0.00561 × x) 1.755 × 10^6 × sin(0.00561 × x) + 3.556 × 10^5 × cos(0.006732 × x) -1.409 × 10^5 × sin(0.006732 × x) - 2.893 × 10^4 × cos(0.007854 × x) -2385 × sin(0.007854 × x)			
Fitted line E_{C0X}	1.3238e× 10^{-4} × x + 5.0795 × 10^{-4}			

5.4.3 Determining part location based on error models

Based on the error models presented in Tables 5.5 and 5.6, error values for each part location can be determined. A test part containing two slots has been selected to realise the theory mentioned in Sections 5.4.1 and 5.4.2. Two 10 mm slots are sufficient to capture squareness and straightness errors in XY plane. Technical drawing of this part has been shown in Figure 5-14. In order to estimate the error of tool tip in each position, a dummy tool path has been selected. The tool path has been presented in Figure 5-15 and attached in Appendix H. 23 points in the tool path has been selected to calculate the total error to finish the job.

The tool path shown in Figure 5-15 has been simulated on six different locations on the bed to find the area which has the least error. The error values calculated based on the formulas provided in Sections 5.4.1 and 5.4.2. Also, The total error of machining has been calculated based on the following formulas:

$$\text{Error for each point} = \sqrt{e_x^2 + e_y^2} \tag{5.29}$$

$$\text{Total error} = \sum_{i=1}^{23} \sqrt{e_{x_i}^2 + e_{y_i}^2} \tag{5.30}$$

Tables 5.7, 5.8, 5.9, 5.10, 5.11 and 5.12 list all the calculated errors for each simulated machining tool path.

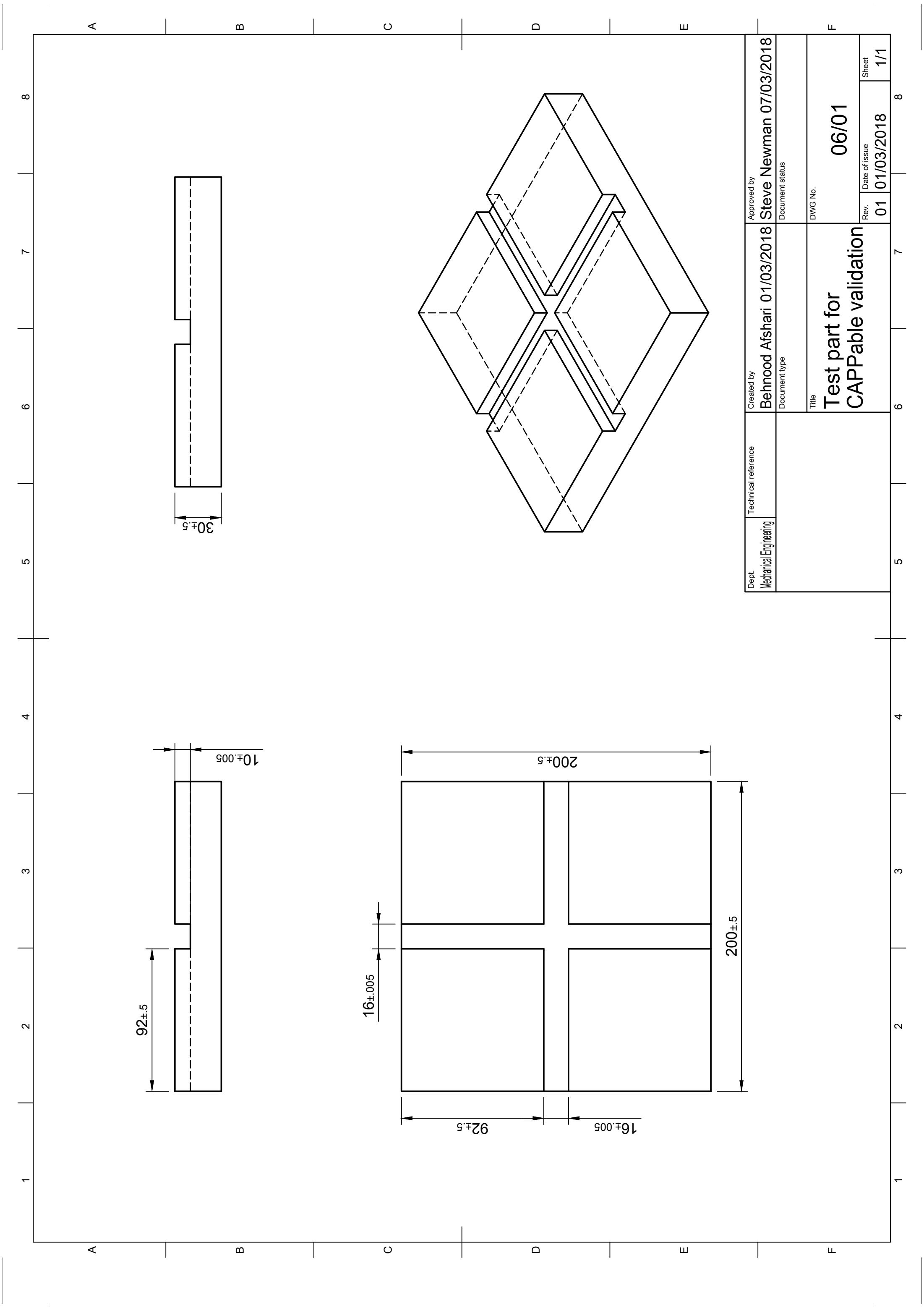


Figure 5-14: Technical drawing of the test part used for experiment

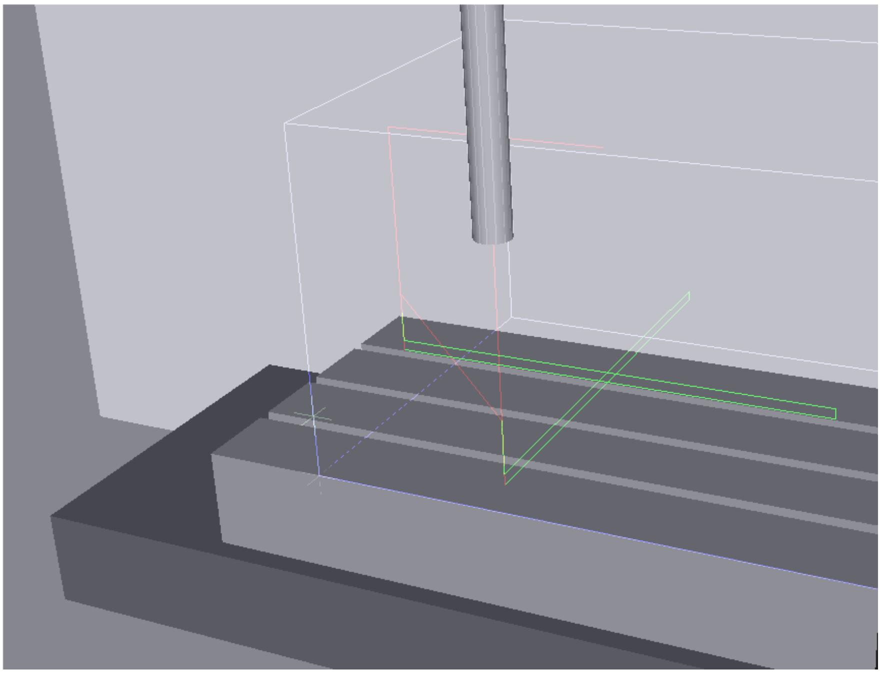


Figure 5-15: Tool path selected to validate CAPPable

It can be predicted from the simulation results that the 4th setup point has the minimum squareness and straightness errors. Section 5.4.4 will demonstrate the application of the theory outlined previously followed by a machining case.

5.4.4 Empirical validation of CAPPable

A profile of the Dugard machine tool containing sufficient information for part optimisation has been generated. This profile has been listed in Appendix I. All the data which can be collected by the ball bar test has been included in this profile, such as straightness and squareness errors in X and Y directions. The EXPRESS structure listed in Table 5.13 has been used to store data for the ball bar test results. Based on the theory described in 6.2.3, CAPPable has determined the best part location at the 4th setup point.

Table 5.7: Simulation of errors generated from a dummy tool path

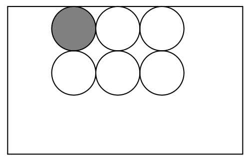
Test	X_i	Y_i	E_{XY} (µm)	E_{C0X} (rad)	e_x (µm)	E_{YX} (µm)	e_y (µm)	Total Error (µm)	Location of test
	-300	0	1.6	-0.000685	1.6	0.0	0.0	2.6	
	-300	50	0.4	-0.000685	0.4	0.0	0.0	0.2	
	-300	100	0.0	-0.000685	0.0	0.0	0.0	0.0	
	-250	100	0.0	-0.000569	0.0	0.4	0.4	0.1	
	-200	100	0.0	-0.000452	0.0	1.1	1.1	1.2	
	-150	100	0.0	-0.000335	0.0	1.3	1.3	1.7	
	-100	100	0.0	-0.000219	0.0	1.2	1.2	1.4	
	-150	100	0.0	-0.000335	0.0	1.3	1.3	1.7	
	-200	100	0.0	-0.000452	0.0	1.1	1.1	1.2	
	-250	100	0.0	-0.000569	0.0	0.4	0.4	0.1	
	-300	100	0.0	-0.000685	0.0	0.0	0.0	0.0	
	-265	135	0.2	-0.000604	0.2	0.2	0.2	0.1	
1	-230	170	0.9	-0.000522	0.9	0.7	0.7	1.2	
	-195	205	1.9	-0.000440	1.9	1.2	1.2	5.1	
	-200	200	1.8	-0.000452	1.8	1.1	1.1	4.5	
	-200	150	0.4	-0.000452	0.4	1.1	1.1	1.4	
	-200	100	0.0	-0.000452	0.0	1.1	1.1	1.2	
	-200	50	0.4	-0.000452	0.4	1.1	1.1	1.4	
	-200	0	1.6	-0.000452	1.6	1.1	1.1	3.8	
	-200	50	0.4	-0.000452	0.4	1.1	1.1	1.4	
	-200	100	0.0	-0.000452	0.0	1.1	1.1	1.2	
	-200	150	0.4	-0.000452	0.4	1.1	1.1	1.4	
	-200	200	1.8	-0.000452	1.8	1.1	1.1	4.5	
								Total = 6.1	

Table 5.8: Simulation of errors generated from a dummy tool path

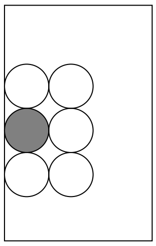
Test	X_i	Y_i	E_{XY} (µm)	E_{C0X} (rad)	e_x (µm)	E_{YX} (µm)	e_y (µm)	Total Error (µm)	Location of test
	-100	0	2.2	-0.000219	2.2	1.2	1.2	6.3	
	-100	50	0.6	-0.000219	0.6	1.2	1.2	1.8	
	-100	100	0.0	-0.000219	0.0	1.2	1.2	1.4	
	-50	100	0.0	-0.000102	0.0	1.2	1.2	1.4	
	0	100	0.0	0.000015	0.0	1.2	1.2	1.7	
	50	100	0.0	0.000131	0.0	1.2	1.2	1.4	
	100	100	0.0	0.000248	0.0	1.2	1.2	1.4	
	50	100	0.0	0.000131	0.0	1.2	1.2	1.4	
	0	100	0.0	0.000015	0.0	1.3	1.3	1.7	
	-50	100	0.0	-0.000102	0.0	1.2	1.2	1.4	
	-100	100	0.0	-0.000219	0.0	1.2	1.2	1.4	
	-65	135	0.0	-0.000137	0.0	1.2	1.2	1.4	
2	-30	170	1.2	-0.000055	1.2	1.3	1.3	3.1	
	5	205	2.6	0.000026	2.6	1.3	1.3	8.4	
	0	200	2.4	0.000015	2.4	1.3	1.3	7.4	
	0	150	0.6	0.000015	0.6	1.3	1.3	2.1	
	0	100	0.0	0.000015	0.0	1.3	1.3	1.7	
	0	50	0.6	0.000015	0.6	1.3	1.3	2.0	
	0	0	2.2	0.000015	2.2	1.3	1.3	6.5	
	0	50	0.6	0.000015	0.6	1.3	1.3	2.0	
	0	100	0.0	0.000015	0.0	1.3	1.3	1.7	
	0	150	0.6	0.000015	0.6	1.3	1.3	2.1	
	0	200	2.4	0.000015	2.4	1.3	1.3	7.4	
								Total = 8.2	

Table 5.9: Simulation of errors generated from a dummy tool path

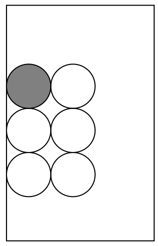
Test	X_i	Y_i	E_{XY} (µm)	E_{C0X} (rad)	e_x (µm)	E_{YX} (µm)	e_y (µm)	Total Error (µm)	Location of test
	100	0	0.2	0.000248	0.2	1.2	1.2	1.5	
	100	50	0.0	0.000248	0.0	1.2	1.2	1.4	
	100	100	0.0	0.000248	0.0	1.2	1.2	1.4	
	150	100	0.0	0.000364	0.0	1.3	1.3	1.7	
	200	100	0.0	0.000481	0.0	1.1	1.1	1.2	
	250	100	0.0	0.000598	0.0	0.4	0.4	0.1	
	300	100	0.0	0.000714	0.0	0.0	0.0	0.0	
	250	100	0.0	0.000598	0.0	0.4	0.4	0.1	
	200	100	0.0	0.000481	0.0	1.1	1.1	1.2	
	150	100	0.0	0.000364	0.0	1.3	1.3	1.7	
	100	100	0.0	0.000248	0.0	1.2	1.2	1.4	
	135	135	0.0	0.000329	0.0	1.3	1.3	1.7	
3	170	170	0.1	0.000411	0.1	1.3	1.3	1.7	
	205	205	0.2	0.000493	0.2	1.0	1.0	1.0	
	200	200	0.2	0.000481	0.2	1.1	1.1	1.2	
	200	150	0.1	0.000481	0.1	1.1	1.1	1.2	
	200	100	0.0	0.000481	0.0	1.1	1.1	1.2	
	200	50	0.0	0.000481	0.0	1.1	1.1	1.2	
	200	0	0.2	0.000481	0.2	1.1	1.1	1.2	
	200	50	0.0	0.000481	0.0	1.1	1.1	1.2	
	200	100	0.0	0.000481	0.0	1.1	1.1	1.2	
	200	150	0.1	0.000481	0.1	1.1	1.1	1.2	
	200	200	0.2	0.000481	0.2	1.1	1.1	1.2	
								Total = 5.2	

Table 5.10: Simulation of errors generated from a dummy tool path

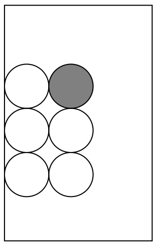
Test	X_i	Y_i	E_{XY} (µm)	E_{C0X} (rad)	e_x (µm)	E_{YX} (µm)	e_y (µm)	Total Error (µm)	Location of test
	100	-200	0.2	0.000240	0.2	-0.8	-0.8	0.7	
	100	-150	0.1	0.000240	0.1	-0.8	-0.8	0.7	
	100	-100	0.1	0.000240	0.1	-0.8	-0.8	0.7	
	150	-100	0.1	0.000355	0.1	-0.5	-0.5	0.3	
	200	-100	0.1	0.000471	0.1	-0.2	-0.2	0.0	
	250	-100	0.1	0.000586	0.1	-0.2	-0.2	0.1	
	300	-100	0.1	0.000702	0.1	-0.4	-0.4	0.2	
	250	-100	0.1	0.000586	0.1	-0.2	-0.2	0.1	
	200	-100	0.1	0.000471	0.1	-0.2	-0.2	0.0	
	150	-100	0.1	0.000355	0.1	-0.5	-0.5	0.3	
	100	-100	0.1	0.000240	0.1	-0.8	-0.8	0.7	
	135	-65	0.1	0.000321	0.1	-0.7	-0.7	0.5	
4	170	-30	0.2	0.000402	0.2	-0.4	-0.4	0.2	
	205	5	0.2	0.000483	0.2	-0.2	-0.2	0.1	
	200	0	0.2	0.000471	0.2	-0.2	-0.2	0.1	
	200	-50	0.1	0.000471	0.1	-0.2	-0.2	0.0	
	200	-100	0.1	0.000471	0.1	-0.2	-0.2	0.0	
	200	-150	0.1	0.000471	0.1	-0.2	-0.2	0.0	
	200	-200	0.2	0.000471	0.2	-0.2	-0.2	0.1	
	200	-150	0.1	0.000471	0.1	-0.2	-0.2	0.0	
	200	-100	0.1	0.000471	0.1	-0.2	-0.2	0.0	
	200	-50	0.1	0.000471	0.1	-0.2	-0.2	0.0	
	200	0	0.2	0.000471	0.2	-0.2	-0.2	0.1	
								Total = 2.2	

Table 5.11: Simulation of errors generated from a dummy tool path

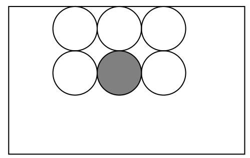
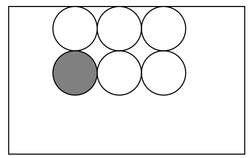
Test	X_i	Y_i	E_{XY} (µm)	E_{C0X} (rad)	e_x (µm)	E_{YX} (µm)	e_y (µm)	Total Error (µm)	Location of test
	-100	-200	2.5	-0.000222	2.5	0.8	0.8	6.9	
	-100	-150	1.2	-0.000222	1.2	0.8	0.8	2.1	
	-100	-100	0.8	-0.000222	0.8	0.8	0.8	1.2	
	-50	-100	0.8	-0.000107	0.8	0.5	0.5	0.9	
	0	-100	0.8	0.000009	0.8	0.0	0.0	0.6	
	50	-100	0.8	0.000124	0.8	-0.6	-0.6	1.0	
	100	-100	0.8	0.000240	0.8	-0.8	-0.8	1.3	
	50	-100	0.8	0.000124	0.8	-0.6	-0.6	1.0	
	0	-100	0.8	0.000009	0.8	0.0	0.0	0.6	
	-50	-100	0.8	-0.000107	0.8	0.5	0.5	0.9	
	-100	-100	0.8	-0.000222	0.8	0.8	0.8	1.2	
	-65	-65	1.0	-0.000141	1.0	0.6	0.6	1.4	
5	-30	-30	1.7	-0.000060	1.7	0.3	0.3	3.0	
	5	5	2.2	0.000020	2.2	-0.1	-0.1	4.8	
	0	0	2.2	0.000009	2.2	0.0	0.0	4.8	
	0	-50	1.2	0.000009	1.2	0.0	0.0	1.4	
	0	-100	0.8	0.000009	0.8	0.0	0.0	0.6	
	0	-150	1.2	0.000009	1.2	0.0	0.0	1.4	
	0	-200	2.5	0.000009	2.5	0.0	0.0	6.3	
	0	-150	1.2	0.000009	1.2	0.0	0.0	1.4	
	0	-100	0.8	0.0000091	0.8	0.0	0	0.6	
	0	-50	1.2	0.000009	1.2	0.0	0.0	1.4	
	0	0	2.2	0.000009	2.2	0.0	0.0	4.8	
								Total = 7.1	

Table 5.12: Simulation of errors generated from a dummy tool path

Test	X_i	Y_i	E_{XY} (µm)	E_{C0X} (rad)	e_x (µm)	E_{YX} (µm)	e_y (µm)	Total Error (µm)	Location of test
	-300	-200	1.8	-0.000684	1.8	0.0	0.0	3.2	
	-300	-150	0.6	-0.000684	0.6	0.0	0.0	0.4	
	-300	-100	0.2	-0.000684	0.2	0.0	0.0	0.0	
	-250	-100	0.2	-0.000569	0.2	0.2	0.2	0.1	
	-200	-100	0.2	-0.000453	0.2	0.6	0.6	0.4	
	-150	-100	0.2	-0.000338	0.2	0.8	0.8	0.7	
	-100	-100	0.2	-0.000222	0.2	0.8	0.8	0.7	
	-150	-100	0.2	-0.000338	0.2	0.8	0.8	0.7	
	-200	-100	0.2	-0.000453	0.2	0.6	0.6	0.4	
	-250	-100	0.2	-0.000569	0.2	0.2	0.2	0.1	
	-300	-100	0.2	-0.000684	0.2	0.0	0.0	0.0	
	-265	-65	0.4	-0.000603	0.4	0.1	0.1	0.2	
6	-230	-30	1.0	-0.000523	1.0	0.3	0.3	1.1	
	-195	5	1.6	-0.000442	1.6	0.6	0.6	2.9	
	-200	0	1.6	-0.000453	1.6	0.6	0.6	2.9	
	-200	-50	0.6	-0.000453	0.6	0.6	0.6	0.6	
	-200	-100	0.2	-0.000453	0.2	0 .6	0.6	0.4	
	-200	-150	0.6	-0.000453	0.6	0.6	0.6	0.7	
	-200	-200	1.8	-0.000453	1.8	0.6	0.6	3.6	
	-200	-150	0.6	-0.000453	0.6	0.6	0.6	0.7	
	-200	-100	0.2	-0.0000453	0.2	0.6	0.6	0.4	
	-200	-50	0.6	-0.000453	0.6	0.6	0.6	0.6	
	-200	0	1.6	-0.000453	1.6	0.6	0.6	2.9	
								Total = 4.9	

To validate CAPPable, the part presented in Figure 5-14 has been machined 6 times in six different locations on the Dugard bed. Machined parts have been measured on a Brown & Sharpe CMM retrofitted with Renishaw probe. Figure 5-16 shows all the points scanned with the CMM, and the orientation of the first test.
Next, the deviation of each data point from the best fitted line has been calculated in X and Y directions. The total error has been determined using the Equations 5.31 and 5.32. To predict the squareness errors of the machined slots, deviation of these slots from 90 degree has been calculated using the Equation 5.33.

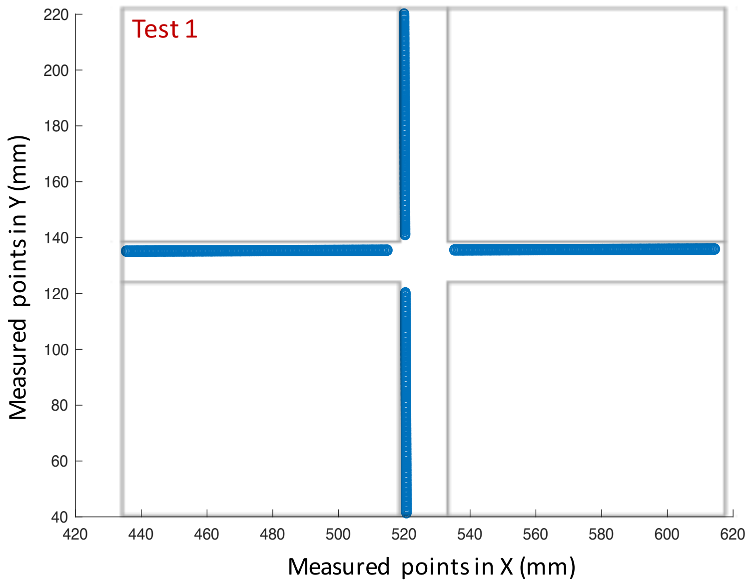


Figure 5-16: CMM readings plot and data points regression lines

$$E_{XY_{tm}} = \sum_{i=1}^{n} \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}} \tag{5.31}$$

$$E_{YX_{tm}} = \sum_{i=1}^{n} \sqrt{\frac{\sum_{i=1}^{n} (x_i - \hat{x}_i)^2}{n}} \tag{5.32}$$

$$E_{C0X_m} = \arctan\left(\frac{m_1 + m_1}{1 - m_1 \times m_2}\right) \tag{5.33}$$

Where, y_i and x_i are the actual data points measured by CMM, $\hat{y}_i$ and $\hat{x}_i$ are the data obtained from the best fitted line, n is number of measurement points, and m_1 and m_2 are the gradients of the fitted lines in X and Y. $E_{XY_{tm}}$ and $E_{YX_{tm}}$ are the total estimated straightness values and E_{C0X_m} is the estimated squareness value obtained from the part. E_{XY_m} and E_{YX_m} are the fitted lines representing straightness values in

Table 5.13: Data structure for the ball bar test

Label	Data model description
Axes capability	ENTITY axes_capability SUBTYPE OF(machining_capability); its_test_method : test_method; its_axes_accuracy : SET [0:?] OF axes_accuracy; END_ENTITY;
Axes accuracy	ENTITY axes_accuracy; setup_location_x : length_measure; setup_location_y : length_measure; its_movement_along_x : movement_along_x; its_movement_along_y : movement_along_y; its_movement_along_z : movement_along_z; plane_xy_squareness : plane_angle_measure; plane_yz_squareness : plane_angle_measure; plane_xz_squareness : plane_angle_measure; END_ENTITY;
Movement along X	ENTITY movement_along_x; its_linear_xx_error : length_measure; its_angular_error : plane_angle_measure; its_straightness_yx_error : length_measure; its_straightness_zx_error : length_measure; END_ENTITY;
Movement along Y	ENTITY movement_along_y; its_linear_yy_error : length_measure; its_angular_error : plane_angle_measure; its_straightness_xy_error : length_measure; its_straightness_zy_error : length_measure; END_ENTITY;
Movement along Z	ENTITY movement_along_z; its_linear_zz_error : length_measure; its_angular_error : plane_angle_measure; its_straightness_xz_error : length_measure; its_straightness_yz_error : length_measure; END_ENTITY;
Test method	TYPE test_method=ENUMERATION OF (STRAIGHTEDGE, TELESCOPE, TAUT_WIRE, LASER_INTERFEROMETER, BALL_BAR); END_TYPE;

X and Y directions. Table 5.14 lists the regression lines for the measured data points in Figure 5-16 and the calculated straightness and squareness values using Equations 5.31 and 5.33. It can be seen from Table 5.14 that the measured error values from the machined part also shows better accuracy in setup point 4. The setup point positions have been presented in Figure 5-13.

An additional finding from this test is that the gradients of fitted lines in X and Y directions are almost the same for the six machined parts. This agrees with the findings of the ball bar test which found similar squareness errors in each of the eight testing locations.

5.5 Summary

To evaluate the fitness of the proposed framework, three separate test cases have been presented. First, CAPPable is used to select the capable machining resources based on their availability. This assessment takes place in the macro process planning stage of part manufacturing. The generated profiles of machines is compared against machining requirements, such as the working envelope and the tool availability.

Next, the micro process planning stage has been selected to implement CAPPable. The actual condition of the PKM legs has been captured and incorporated with the proposed framework. Generated MCP file has been used to find the best location of a workpiece on the PKM bed. The optimisation problem has been formulated considering the error factors in the parallel legs.

In Section 5.4, the ball bar test has been used to model the straightness and squareness errors in the Dugard machine tool listed in Appendix I. The error data has been included in the profile of the Dugard machine tool. Thus, CAPPable can perform capability check to obtain the best part location on the machine bed. After determining the best part location using error models, the machined parts have been measured on the CMM machine. The measured data on CMM agrees with the results obtained from CAPPable.

Table 5.14: Measured error values from the experiment

Test part 1	Best fitted lines based on CMM data	$E_{YX_m} = 0.0042x + 133.3373$ $E_{XY_m} = -0.0041y + 520.8094$
	Caculated error values from measurement	$E_{C0X_m} = 0.0057$ $E_{XY_{tm}} = 0.0051$ $E_{YX_{tm}} = 0.0063$
Test part 2	Best fitted lines based on CMM data	$E_{YX_m} = 0.003x + 134.3487$ $E_{XY_m} = -0.0028y + 519.7123$
	Caculated error values from measurement	$E_{C0X_m} = 0.011$ $E_{XY_{tm}} = 0.0043$ $E_{YX_{tm}} = 0.0178$
Test part 3	Best fitted lines based on CMM data	$E_{YX_m} = -0.004x + 137.1753$ $E_{XY_m} = -0.0042y + 519.5739$
	Caculated error values from measurement	$E_{C0X_m} = 0.011$ $E_{XY_{tm}} = 0.0076$ $E_{YX_{tm}} = 0.0128$
Test part 4	Best fitted lines based on CMM data	$E_{YX_m} = -0.0047x + 137.4148$ $E_{XY_m} = 0.0048y + 497.2025$
	Caculated error values from measurement	$E_{C0X_m} = 0.0057$ $E_{XY_{tm}} = 0.0042$ $E_{YX_{tm}} = 0.0026$
Test part 5	Best fitted lines based on CMM data	$E_{YX_m} = -0.0086x + 139.1638$ $E_{XY_m} = 0.0088y + 497.0145$
	Caculated error values from measurement	$E_{C0X_m} = 0.011$ $E_{XY_{tm}} = 0.0037$ $E_{YX_{tm}} = 0.0098$
Test part 6	Best fitted lines based on CMM data	$E_{YX_m} = 0.0044x + 133.8313$ $E_{XY_m} = -0.0042y + 497.7410$
	Caculated error values from measurement	$E_{C0X_m} = 0.011$ $E_{XY_{tm}} = 0.0034$ $E_{YX_{tm}} = 0.0092$

Chapter 6

Discussion

6.1 Introduction

This chapter discusses the research presented in this thesis, highlighting novel aspects of the system proposed. Thoughts on the work conducted will be offered, and mapped against the scope provided in Chapter 1. Finally advantages and limitations of the CAPPable framework will be discussed.

The research presented in this thesis investigates the development of a process planning system to manufacture parts based on the status of available machining resources. Machine tools normally come in different designs and configurations. In order to capture the real capability of machine tools, existing machine tool models have been reviewed. Two levels of manufacturing process planning have been proposed to implement CAPPable, micro level and macro level. Macro level CAPPable has been used to determine the possibility of manufacturing a part on various CNC machines. Micro level CAPPable has been used to determine the part location based on the geometric errors of a machine tool which will result in the least amount of straightness and squareness errors.

6.2 State-of-the-art in methods for capturing machining capability

A review of the literature on the state-of-the-art process planning methods for CNC milling was performed in Chapter 2. A large number of CAPP systems were identified which focused on generating machineable process plans. Most of these CAPP systems are designed based on the assumption that machine tools are performing without any degradation (e.g. tool wear). Machining resources are generally ageing over their operational life which can limit the application and accuracy of these CAPP systems. There are only a few papers that emphasise the available machine tool models to generate reliable process plans based on the actual machine status. A process planning system can only ensure that the part program is machinable if it works based on machine tool models. Also, the reasoning system used in this assessment should be capable enough to incorporate machining requirements into the decision-making system. This has been fulfilled with the emergence of STEP-NC standards. STEP-NC provides end-users with the list of machining requirements which can be used for decision-making purposes.

6.3 A Novel Framework for capturing the capability of machine tools

The framework outlined in Chapter 3 provides a novel method for capturing the real capability of machines. The main concept of the framework is to have an intelligent process planning system which can decide whether the machine is capable of handling a specific job. The proposed framework contains a time-sensitive profile of the machine tool health. This profile includes a machine tool model which can be used for comparative tests across various machines on the shop-floor.

The framework consists of two main parts:

1. The CAPPable framework which performs the main decision-making without any human intervention. This system is capable of reading an MCP file together with machining instructions. The output of the system identifies whether or not it is feasible to manufacture a part on a specific machine.

2. The MCP file which stores the most recent condition of the machine. This file contains all of the possible machining configurations which can be fed into CAPPable.

The proposed framework is an improvement over the currently used process planning systems. This is mainly due to the fact that available process planning systems work based on the assumption that degraded CNCs will still result in the final product being within tolerance. However, there is a possibility of rejects when there are various machine tools on the shop-floor with different capabilities. At the moment, the machineability of a part is assessed by an experienced process planner who has knowledge about the available machining resources. This knowledge is obtained from his past experience of working with the same machine tools.

6.4 Implementation of CAPPable prototype

In order to implement CAPPable, a prototype has been generated using the Java platform. The application software has been developed previously as a part of an EU project titled 'STEPMAN'. This application is capable of generating MCP and STEP-NC files. The MCP generator has been adapted to transfer the machine tool health parameters to the CAPPable framework.

6.5 Macro level process planning test cases

In order to determine the capabilities of the framework discussed in Chapter 3, a prototype implementation was developed in the form of a macro process planning system. The entire decision-making program was developed using the Java programming language and developed in an object-oriented structure. The prototype is capable of receiving a part program and an MCP file in the STEP-NC format. The MCP file contains the latest status of the machine, and the part program includes the design tolerances and machining working steps. The output of the proposed system helps the process planner to check the machineability of a part in the early stage of process planning.

6.6 Optimising the part location on the PKM bed

The optimal position of a sample part has been determined based on the capability of the 'Equator' parallel legs. The sample part has been virtually located on different positions on the PKM bed. Minimum translational error along the parallel legs have been considered to find the best location to setup the part. Based on the information retrieved from the 'Equator' MCP file, the actual tool tip positions have been calculated along a tool path. The same tool path has been simulated in different locations on the PKM bed. The optimal setup location has been determined using the PSO algorithm. The results show that locating the workpiece with the fully extended legs can increase the chance of error. Thus, suggested workpiece location for this test case is near to centre of the bed where the parallel legs have closer distance to the workpiece.

6.7 Optimising the part location on the Dugard machine tool

The geometric errors of the Dugard machine tool has been captured using a Renishaw ball bar device. The recorded values have been stored in the Dugard MCP file. Using the kinematic model of the Dugard machine tool, and the decision engine provided by CAPPable, the best setup point for a sample part has been determined. The solution which was provided by CAPPable has been validated by machining 6 parts on the Dugard bed, and measuring the accuracy of the parts on a CMM machine.

6.8 Advantages and limitations of the CAPPable system

The main advantages of using the CAPPable framework are as follows:

1. It offers process planners the ability to check the quality of a generated process plan before any capability issues will affect the production. This can be considered as a quality check on the generated process plans.

2. The quality of a process plan relies heavily on the knowledge of an experienced process planner. This is due to the continuously gained knowledge of working with the same manufacturing resources on a daily basis. When skilled process planner leaves a firm, this valuable knowledge will be lost. CAPPable offers the ability to save this knowledge and share it with the same or other departments. It also can greatly increase the rate at which new process planners can learn about the capability that their machines can offer.

3. Capability profiles can be extended to work with any types of machines including custom made machines and hybrid parallel kinematic robots. The method of capturing the capability of these machines can be performed online using sensors, or offline. Capturing offline data requires a reliable kinematic model of the machines.

The following is a list that discusses the limitations of the framework proposed in this research:

1. The output of the CAPPable highly relies on the data capturing method used to store the capability of the machines. The accuracy of CAPPable cannot exceed that of the method used in measuring the capability of the machine.

2. The CAPPable only considers errors which are measured and provided. For exam-

ple in Section 5.4 the squareness and straightness errors of the Dugard machine tool were measured and provided. Although in practice there are more errors which can be present which can affect the machining accuracy or alter the actual capability of a machine. Therefore the more errors which are provided into the CAPPable framework, the more likely CAPPable is able to provide an accurate assessment of the available machining capabilities.

Chapter 7

Conclusions and future work

7.1 Introduction

This chapter will cover the conclusions obtained from this research along with the overall contribution to knowledge. Potential areas in which this research could be expanded and further investigated is covered in the final section of this chapter.

7.2 Conclusions

The conclusions resulting from this research are as follows:

1. Machine tools are available in different configurations and capabilities in industry. Process planners rely on their experience and knowledge of these manufacturing resources to maintain the quality of a manufactured part. This approach is extremely time-consuming and the results vary based on the process planner's experience. As a result, various CAPP systems have emerged to facilitate this decision-making. However, none of these systems generate process plans which adhere to the status of the available machining resources.

2. It is possible to generate process plans based on the real capability of a machine. This can be done by developing a machine tool capability profile and linking it with a decision-making engine. The machining capability profile can represent the current status of the machine at that stage of its life. By providing machine tool health parameters, the manufacturability of a part can be assessed.

3. A 3-axis machine tool will contain: three axes, a variety of cutting tools and a cuboid shape working envelope, however, an advanced machine tool will contain far more complex parameters for capability assessment. The process planning framework used in this research is extensible, and can be linked with more manufacturing resources. The proposed framework has been tested on a PKM and Dugard machine tool, and focused on the kinematic structure of machines. There are more health parameters required to add to the current model to make sure that the manufactured part adheres to the design. The objected-oriented design of CAPPable can be a solution to this issue. This system can easily be adapted and expanded to account for further developments in the field of manufacturability assessment. Each health parameter can be developed as a separate module for the CAPPable framework.

4. The prototype's ability was shown to work with the following PKM machine capabilities: frame size, leg length, plate dimension and translational errors. The prototype was able to compute the best part location on a PKM bed with the use of a particle swarm optimisation.

5. The prototype's ability was also shown to work with the following 3-axis milling machine capabilities: straightness errors, squareness errors, available cutting tools, machining envelope and setup points.

6. Geometric errors affecting machining quality can be avoided using CAPPable. geometric errors such as straightness and squareness values can be stored in MCP files, and incorporated with the kinematic model of machines to determine the optimal part location. This has been investigated in Section 5.4 which shows that the hypothesis in Section 1.6 should be accepted.

7.3 Contribution to Knowledge

The main contribution to knowledge of this research lies in the novelty of the framework developed to assess the feasibility of manufacturing a part on a degraded machine tool. The framework allows for process planners to test the availability of manufacturing resources before the machining program execution. Having such a system can be valuable for machining some high-value products, such as turbine blades. At the moment, these turbine blades are manufactured in batches and any inconsistency in production can affect the allocation of machining resources. Discovering these issues in production can be done before machining the actual part by checking the capability of the available machining resources.

Furthermore, the proposed system works based on the STEP-NC standard which can

be extended to represent information about any type of CNC machine such as turning, drilling, grinding and additive manufacturing.
The proposed framework provided in this research can determine the best part location based on the actual machining resources. Process planning based on the real abilities of old machines is enabled using the framework presented in this thesis which is a valuable contribution to this field of research.

7.4 Future work

This work has made a novel contribution to the body of knowledge and it is envisaged that aspects of the proposed framework could be extended further as it has been described in the following sections.

7.4.1 Including additional health parameters

This research focused on capturing the real capabilities of CNC machines using available kinematic models. The first extension of this work could be to add more capability parameters such as fixturing parameters. The overall capability defined in CAPPable only considers the working volume available for a CNC machine. This can be further investigated by considering the available fixturing systems to hold parts of various shapes and sizes when subjected to the external force.

7.4.2 Considering dynamic and thermal errors

Only the geometric errors of a machine have been considered in this thesis, assuming that the machining errors affecting the general health of the machines are straightness and squareness errors. This is not always true in machining operations. Capturing dynamic and thermal errors of machine tools, and applying these types of errors to build more robust machine tool error models can improve the proposed process planning system.

7.4.3 Other machining and manufacturing technologies

Data model presented in this research has only been realised for milling technologies including serial and parallel kinematic machines, but this data model can be further extended for any manufacturing technologies such as wire EDM, CMM and Additive which use different kinematic configurations. Also, the various manufacturing technologies covered in STEP-NC can be included in CAPPable as the data structure is the same.

7.4.4 Using joint capabilities of two machines

Existing machine tools in a shop-floor may offer different level of capabilities. Some renewed or readjusted machine tools may work at a higher accuracy whereas the others may offer additional axes, cutting tools or workpiece holding or fixturing. Based on this information, deciding what is the best combination of machines that can deliver higher quality can be a complex problem. Having access to joint capabilities of the available machine tools in a factory can be a solution to this problem.

7.4.5 Capability assessment enabled by CAM

The current process planning system can be used in conjunction with available CAM software. A CAM software will read a machine tool profile from a database. The imported machine tool data can be utilised to assess the health of machine tool and its latest status. Having such a system can compare available resources with the machining strategy which a process planner may intend to use. Also, there is no need for an experienced process planner as the CAM software enabled by MCP can automate this process.

7.4.6 Smart controllers

Smart controllers can be integrated with the proposed planning system, to accept different machining codes, and assess them based on the latest status of the machine. This can be used as a validation tool which is offered by the machine tool manufacturer.

7.4.7 Cloud-based capability analysis

Digital manufacturing introduced by Industry 4.0 has been recognised as the future trend in manufacturing. To realise this concept, the MCP based process planning framework developed in this research could be introduced into a cloud-based system. The machining capability profiles can be stored inside a the cloud system and can be utilised by process planners to work within that system (Mourad et al., 2016). Also, external customers can retrieve the latest capabilities of a company across this cloud-based capability to make sure that the company can adhere to the manufacturing requirements needed by the product.

References

1. Agapiou, J. S., Gu, J., and Kurgin, S. (2016). Global offset compensation for cnc machine tools based on workpiece errors. *Procedia Manufacturing*, 5:442–454.

2. Allen, R., Harding, J., and Newman, S. (2005). The application of step-nc using agent-based process planning. *International Journal of Production Research*, 43(4):655–670.

3. Alting, L. and Zhang, H. (1989). Computer aided process planning: the state-of-the-art survey. *International Journal of Production Research*, 27(4):553–585.

4. Altintas, Y., Verl, A., Brecher, C., Uriarte, L., and Pritschow, G. (2011). Machine tool feed drives. *CIRP Annals-Manufacturing Technology*, 60(2):779–796.

5. Anjum, N., Harding, J., Young, R., Case, K., Usman, Z., and Changoora, T. (2013). Verification of knowledge shared across design and manufacture using a foundation ontology. *International Journal of Production Research*, 51(22):6534–6552.

6. ASME B5.54 (2005). Methods for performance evaluation of computer numerically controlled machining centers. *New York: The American Society of Mechanical Engineers.*

7. Azab, A. (2016). Quadratic assignment problem mathematical modelling for process planning. *International Journal of Computer Integrated Manufacturing*, 29(5):561–580.

8. Babic, B., Nesic, N., and Miljkovic, Z. (2008). A review of automated feature recognition with rule-based pattern recognition. *Computers in Industry*, 59(4):321–337.

9. Baheti, R. and Gill, H. (2011). Cyber-physical systems. *The impact of control technology*, 12:161–166.

10. Bierwirth, C. and Mattfeld, D. C. (1999). Production scheduling and rescheduling with genetic algorithms. *Evolutionary computation*, 7(1):1–17.

11. Boer, C., Petitti, M., Lombardi, F., and Simon, J.-P. (1990). A capp/cam expert system for a high productivity, high flexibility cnc turning center. *CIRP Annals-Manufacturing Technology*, 39(1):481–483.

12. Brecher, C., Esser, M., and Witt, S. (2009). Interaction of manufacturing process and machine tool. *CIRP Annals-Manufacturing Technology*, 58(2):588–607.

13. Brunnermeier, S. B. and Martin, S. A. (1999). Interoperability cost analysis of the us automotive supply chain.

14. Bryan, J. (1990). International status of thermal error research. *CIRP Annals-Manufacturing Technology*, 39(2):645–656.

15. Cay, F. and Chassapis, C. (1997). An IT view on perspectives of computer aided process planning research. *Computers in Industry*, 34(3):307–337.

16. Chang, T. C. and Wysk, R. A. (1984). *An introduction to automated process planning systems*. Prentice Hall Professional Technical Reference.

17. Chen, D., Heyer, S., Ibbotson, S., Salonitis, K., Steingrímsson, J. G., and Thiede, S. (2015). Direct digital manufacturing: definition, evolution, and sustainability implications. *Journal of Cleaner Production*, 107:615–625.

18. Choi, Y.-C. and Xirouchakis, P. (2014). A production planning in highly automated manufacturing system considering multiple process plans with different energy requirements. *The International Journal of Advanced Manufacturing Technology*, 70(5-8):853–867.

19. Chryssolouris, G. (2006). *Manufacturing systems: theory and practice*. Springer Science & Business Media.

20. Darbyshire, I. and Davies, B. (1984). Excap: an expert generative process planning system. In *Report on Workshop on Expert Systems in Manufacturing Engineering*. Science and Engineering Research Council, ACME Directorate Swindon, UK.

21. Delmia VNC (2017). Accessed 26 feb 2017. *https://www.3ds.com/products-services/delmia/*.

22. Denavit, J. and Hartenberg, R. S. (1964). *Kinematic synthesis of linkages*. McGraw-Hill.

23. DIN 66301 (1987). Format for the exchange of geometrical information. *Deutsches Institut for Normung*.

24. ElMaraghy, H. and Nassehi, A. (2014). Computer-aided process planning. In *CIRP Encyclopedia of Production Engineering*, pages 266–271. Springer.

25. ElMaraghy, H. A. (1993). Evolution and future perspectives of CAPP. *CIRP Annals-Manufacturing Technology*, 42(2):739–751.

26. ElMaraghy, H. A. (2007). Reconfigurable process plans for responsive manufacturing systems. In *Digital enterprise technology*, pages 35–44. Springer.

27. Fernandes, K. J. and Raja, V. H. (2000). Incorporated tool selection system using object technology. *International Journal of Machine Tools and Manufacture*, 40(11):1547–1555.

28. Fesperman, R. R., Moylan, S. P., Vogl, G. W., and Donmez, M. A. (2015). Reconfigurable data driven virtual machine tool: Geometric error modeling and evaluation. *CIRP Journal of Manufacturing Science and Technology*, 10:120–130.

29. Flynn, J., Vichare, P., Shokrani Chaharsooghi, A., Dhokia, V., and Newman, S. (2015). Single setup ballbar testing of 5-axis machine tools to identify position-independent geometric errors. *The International Conference on Flexible Automation and Intelligent Manufacturin.*

30. Fu, G., Fu, J., Xu, Y., Chen, Z., and Lai, J. (2015). Accuracy enhancement of five-axis machine tool based on differential motion matrix: geometric error modeling, identification and compensation. *International Journal of Machine Tools and Manufacture*, 89:170–181.

31. GB/T 17421 (2003). Test code for machine tools. *National Standardization Technical Committee of metal cutting machine tools.*

32. Gen, M. and Lin, L. (2014). Multiobjective evolutionary algorithm for manufacturing scheduling problems: state-of-the-art survey. *Journal of Intelligent Manufacturing*, 25(5):849–866.

33. Gough, V. and Whitehall, S. (1962). Universal tyre test machine. In *Proc. FISITA 9th Int. Technical Congress*, pages 117–137.

34. Guo, Y., Li, W., Mileham, A. R., and Owen, G. W. (2009). Applications of particle swarm optimisation in integrated process planning and scheduling. *Robotics and Computer-Integrated Manufacturing*, 25(2):280–288.

35. Gupta, M. K., Sood, P., and Sharma, V. S. (2016). Machining parameters optimization of titanium alloy using response surface methodology and particle swarm optimization under minimum-quantity lubrication environment. *Materials and Manufacturing Processes*, 31(13):1671–1682.

36. Ham, W. K., Kim, C. M., and Park, S. C. (2016). Template-based variant process planning for manned assembly lines. *International Journal of Computer Integrated Manufacturing*, pages 1–12.

37. Hebbal, S. and Mehta, N. (2008). Setup planning for machining the features of prismatic parts. *International Journal of Production Research*, 46(12):3241–3257.

38. Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* MIT press.

39. Horvath, M. (1996). Process planning with genetic algorithms on results of knowledge-based reasoning. *International Journal of Computer Integrated Manufacturing*, 9(2):145–166.

40. Houtzeel, A. (1980). Computer assisted process planning, a first step towards integration. *Proceeding of the CASA/SME Autofact West Conference.*

41. Humphreys, P., McIvor, R., and Huang, G. (2002). An expert system for evaluating the make or buy decision. *Computers & Industrial Engineering*, 42(2-4):567–585.

42. Ibaraki, S., Sawada, M., Matsubara, A., and Matsushita, T. (2010). Machining tests to identify kinematic errors on five-axis machine tools. *Precision Engineering*, 34(3):387–398.

43. ISO 10303-1 (1994). Product data representation and exchange, part 1: Overview and fundamental principles. *International Organization for Standardization.*

44. ISO 10303-105 (1996). Product data representation and exchange, part 105: Integrated application ressource: Kinematics. *International Organization for Standardization.*

45. ISO 10303-11 (1994). Industrial automation systems and integration-product data representation and exchange, part 11: Description methods: The express language reference manual. *International Organization for Standardization.*

46. ISO 10303-21 (2002). Industrial automation systems and integration-product data representation and exchange, part 21: Implementation methods: Clear text encoding of the exchange structure. *International Organization for Standardization.*

47. ISO 10303-22 (1998). Industrial automation systems and integration - product data representation and exchange, part 22: Implementation methods: Standard data access interface. *International Organization for Standardization.*

48. ISO 10303-223 (2008). Industrial automation systems and integration-product data representation and exchange, part 223: Application protocol: Exchange of design and manufacturing product information for cast parts. *International Organization for Standardization.*

49. ISO 14306 (2012). Industrial automation systems and integration - jt file format specification for 3d visualization. *International Organization for Standardization.*

50. ISO 14649-1 (2003). Industrial automation systems and integration physical device control-data model for computerized numerical controllers, part 1: Overview and fundamental principles draft international standard. *International Organization for Standardization.*

51. ISO 14649-10 (2004). Industrial automation systems and integration, data model for computerised controllers, part 10: General process data. *International Organization for Standardization.*

52. ISO 14649-15 (2017). Data model for computerized numerical controllers - part 15: Process data for contour cutting of wood and glass. *International Organization for Standardization.*

53. ISO 14649-17 (2017). Data model for computerized numerical controllers - part 17: Process data for additive manufacturing. *International Organization for Standardization.*

54. ISO 14649-201 (2011). Data model for computerized numerical controllers, part 201: Machine tool data for cutting processes. *International Organization for Standardization.*

55. ISO 230-1 (2012). Test code for machine tools, part 1: Geometric accuracy of machines operating under no-load or quasi-static condition. *International Organization for Standardization.*

56. ISO 6983-1 (1982). Numerical control of machines - program format and definition of address words, part 1: Data format for positioning, line and contouring control systems. *International Organization for Standardization.*

57. ISO 8373 (2012). Robots and robotic devices - vocabulary. *International Organization for Standardization.*

58. Jalaludin, A. H., Shukor, M. H. A., Mardi, N. A., Sarhan, A. A. D. M., Ab Karim, M. S., Besharati, S. R., Badiuzaman, W. N. I. W., and Dambatta, Y. S. (2017). Development and evaluation of the machining performance of a cnc gantry double motion machine tool in different modes. *The International Journal of Advanced Manufacturing Technology*, pages 1–10.

59. Jedrzejewski, J., Kwasny, W., and Modrzycki, W. (2008). Precise model of hsc machining centre for aerospace parts machining. *Journal of Machine Engineering*, 8(3):29–41.

60. Jha, K. and Gurumoorthy, B. (2000). Multiple feature interpretation across domains. *Computers in industry*, 42(1):13–32.

61. JIS B6330 (1980). Test code for performance and accuracy of numerically controlled machine tools.

62. Jurrens, K. K., Fowler, J. E., and Algeo, M. E. (1995). Modeling of manufacturing resource information. *National Institute of Standards and Technology, NISTIR*, 5707.

63. Kafashi, S., Shakeri, M., and Abedini, V. (2012). Automated setup planning in capp: a modified particle swarm optimisation-based approach. *International Journal of Production Research*, 50(15):4127–4140.

64. Kagermann, H., Helbig, J., Hellinger, A., and Wahlster, W. (2013). *Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0: Securing the Future of*

German Manufacturing Industry; Final Report of the Industrie 4.0 Working Group. Forschungsunion.

65. Kaparthi, S. and Suresh, N. C. (1991). A neural network system for shape-based classification and coding of rotational parts. *The International Journal of Production Research*, 29(9):1771–1784.

66. Kaymakci, M., Kilic, Z., and Altintas, Y. (2012). Unified cutting force model for turning, boring, drilling and milling operations. *International Journal of Machine Tools and Manufacture*, 54:34–45.

67. Kendrick, B. A. (2016). *An Adaptive, Comparative Approach to Improve Accuracy of Manufactured Parts*. PhD thesis, University of Bath.

68. Khodaygan, S. (2014). Manufacturing error compensation based on cutting tool location correction in machining processes. *International Journal of Computer Integrated Manufacturing*, 27(11):969–978.

69. Khoshdarregi, M. R., Tappe, S., and Altintas, Y. (2014). Integrated five-axis trajectory shaping and contour error compensation for high-speed cnc machine tools. *IEEE/ASME Transactions on Mechatronics*, 19(6):1859–1871.

70. Kjellberg, T., von Euler-Chelpin, A., Hedlind, M., Lundgren, M., Sivard, G., and Chen, D. (2009). The machine tool model-a core part of the digital factory. *CIRP Annals-Manufacturing Technology*, 58(1):425–428.

71. Knapp, G. and Wang, H. (1989). Inductive learning in computer-aided process planning. In *Proceedings of the 1989 IIE Integrated Systems Conference & Society for Integrated Manufacturing Conference, Atlanta, GA, p3*, pages 98–403.

72. Knapp, G. M. and Wang, H.-P. B. (1992). Acquiring, storing and utilizing process planning knowledge using neural networks. *Journal of Intelligent Manufacturing*, 3(5):333–344.

73. Krishna, A. G. and Rao, K. M. (2006). Simultaneous optimal selection of design and manufacturing tolerances with different stack-up conditions using scatter search. *The International Journal of Advanced Manufacturing Technology*, 30(3-4):328–333.

74. Lam, S. and Wong, T. (2000). Recognition of machining features-a hybrid approach. *International Journal of Production Research*, 38(17):4301–4316.

75. Lamikiz, A., De Lacalle, L. L., Ocerin, O., Díez, D., and Maidagan, E. (2008). The denavit and hartenberg approach applied to evaluate the consequences in the tool tip position of geometrical errors in five-axis milling centres. *The International Journal of Advanced Manufacturing Technology*, 37(1-2):122–139.

76. Lee, J. (1995). Machine performance monitoring and proactive maintenance in computer-integrated manufacturing: review and perspective. *International Journal of computer integrated manufacturing*, 8(5):370–380.

77. Lee, J., Bagheri, B., and Jin, C. (2016). Introduction to cyber manufacturing. *Manufacturing Letters*, 8:11–15.

78. Lei, W., Paung, I., and Yu, C.-C. (2009). Total ballbar dynamic tests for five-axis cnc machine tools. *International Journal of Machine Tools and Manufacture*, 49(6):488–499.

79. Lei, W.-T., Wang, W.-C., and Fang, T.-C. (2014). Ballbar dynamic tests for rotary axes of five-axis cnc machine tools. *International Journal of Machine Tools and Manufacture*, 82:29–41.

80. Li, B. and Melkote, S. N. (1999). Improved workpiece location accuracy through fixture layout optimization. *International Journal of Machine Tools and Manufacture*, 39(6):871–883.

81. Li, L., Fuh, J., Zhang, Y., and Nee, A. (2005). Application of genetic algorithm to computer-aided process planning in distributed manufacturing environments. *Robotics and Computer-Integrated Manufacturing*, 21(6):568–578.

82. Li, X., Gao, L., Shao, X., Zhang, C., and Wang, C. (2010a). Mathematical modeling and evolutionary algorithm-based approach for integrated process planning and scheduling. *Computers & Operations Research*, 37(4):656–667.

83. Li, X., Gao, L., and Wen, X. (2013). Application of an efficient modified particle swarm optimization algorithm for process planning. *The International Journal of Advanced Manufacturing Technology*, 67(5-8):1355–1369.

84. Li, X., Gao, L., Zhang, C., and Shao, X. (2010b). A review on integrated process planning and scheduling. *International Journal of Manufacturing Research*, 5(2):161–180.

85. Lim, T., Corney, J., Ritchie, J., and Clark, D. (2001). Optimizing tool selection. *International Journal of Production Research*, 39(6):1239–1256.

86. Liu, X.-J. and Wang, J. (2014). Parallel kinematics. *Type, Kinematics, and Optimal Design, Springer-Verlag, Berlin Heidelberg.*

87. Loffredo, D. (1999). Fundamentals of step implementation. *STEP Tools, Inc.*

88. LongView Advisors (2010). Collaboration & interoperability market report 2010. `http://www.longviewadvisors.com`. Accessed: 2018-07-15.

89. Marquet, F., Company, O., Krut, S., and Pierrot, F. (2002). Enhancing parallel robots accuracy with redundant sensors. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 4, pages 4114–4119. IEEE.

90. Mascle, C., Lu, H., and Maranzana, R. (2007). Machining process planning using decomposition of delta volume. In *Assembly and Manufacturing, 2007. ISAM'07. IEEE International Symposium on*, pages 106–111. IEEE.

91. Merlet, J.-P. (2006). *Parallel robots*, volume 128. Springer Science & Business Media.

92. Mourad, M., Nassehi, A., and Schaefer, D. (2016). Interoperability as a key enabler for manufacturing in the cloud. *Procedia CIRP*, 52:30–34.

93. Nassehi, A. (2007). *The realisation of CAD/CAM/CNC interoperability in prismatic part manufacturing*. PhD thesis, University of Bath.

94. Nassehi, A., Newman, S., and Allen, R. (2006). The application of multi-agent systems for step-nc computer aided process planning of prismatic components. *International Journal of Machine Tools and Manufacture*, 46(5):559–574.

95. Nassehi, A., Newman, S., Dhokia, V., Zhu, Z., and Asrai, R. I. (2012). Using formal methods to model hybrid manufacturing processes. In *Enabling Manufacturing Competitiveness and Economic Sustainability*, pages 52–56. Springer.

96. Nassehi, A. and Vichare, P. (2009). A STEP-NC compliant methodology for modelling manufacturing resources. In *Advanced design and manufacturing based on STEP*, pages 261–281. Springer.

97. Newman, S. and Nassehi, A. (2009). Machine tool capability profile for intelligent process planning. *CIRP Annals-Manufacturing Technology*, 58(1):421–424.

98. Ni, J. (1997). CNC machine accuracy enhancement through real-time error compensation. *Transaction-American society of mechanical engineering journal of manufacturing science and engineering*, 119:717–725.

99. Niebel, B. W. (1965). Mechanized process selection for planning new designs. *ASME paper*, 737.

100. Northrop, L. and Clements, P. (2001). Software product lines. *URL http://www. sei. cmu. edu/library/assets/Philips*, 4(05).

101. Park, M.-W., Park, B.-T., Rho, H.-M., and Kim, S.-K. (2000). Incremental supervised learning of cutting conditons using the fuzzy artmap neural network. *CIRP Annals-Manufacturing Technology*, 49(1):375–378.

102. Parkinson, S., Longstaff, A. P., Fletcher, S., Crampton, A., and Gregory, P. (2012). Automatic planning for machine tool calibration: A case study. *Expert Systems with Applications*, 39(13):11367–11377.

103. Patil, L. and Pande, S. (2002). An intelligent feature-based process planning system for prismatic parts. *International Journal of Production Research*, 40(17):4431–4447.

104. Postlethwaite, S. (1992). *Electronic based accuracy enhancement of CNC machine tools*. PhD thesis, Huddersfield Polytechnic.

105. Prabhakar, S. and Henderson, M. R. (1992). Automatic form-feature recognition using neural-network-based techniques on boundary representations of solid models. *Computer-Aided Design*, 24(7):381–393.

106. Qiao, L.-H., Yang, Z.-B., and Wang, H.-P. B. (1994). A computer-aided process planning methodology. *Computers in Industry*, 25(1):83–94.

107. Ramesh, R., Mannan, M., and Poo, A. (2000a). Error compensation in machine tools-a review: Part i: geometric, cutting-force induced and fixture-dependent errors. *International Journal of Machine Tools and Manufacture*, 40(9):1235–1256.

108. Ramesh, R., Mannan, M., and Poo, A. (2000b). Error compensation in machine tools-a review: Part ii: thermal errors. *International Journal of Machine Tools and Manufacture*, 40(9):1257–1284.

109. Ramesh, R., Mannan, M., and Poo, A. (2000c). Error compensation in machine tools-a review: Part ii: thermal errors. *International Journal of Machine Tools and Manufacture*, 40(9):1257–1284.

110. Rameshbabu, V. and Shunmugam, M. (2009). Hybrid feature recognition method for setup planning from step ap-203. *Robotics and Computer-Integrated Manufacturing*, 25(2):393–408.

111. Rao, K. V., Murthy, B., and Rao, N. M. (2014). Prediction of cutting tool wear, surface roughness and vibration of work piece in boring of aisi 316 steel with artificial neural network. *Measurement*, 51:63–70.

112. Reddy, S. B., Shunmugam, M., and Narendran, T. (1998). Optimal sub-division of the depth of cut to achieve minimum production cost in multi-pass turning using a genetic algorithm. *Journal of Materials Processing Technology*, 79(1):101–108.

113. Renishaw plc (2018). Accessed 06 sep 2018. *http://resources.renishaw.com/details/*.

114. Renner, G. and Ekárt, A. (2003). Genetic algorithms in computer aided design. *Computer-Aided Design*, 35(8):709–726.

115. Ridwan, F. and Xu, X. (2013). Advanced cnc system with in-process feed-rate optimisation. *Robotics and Computer-Integrated Manufacturing*, 29(3):12–20.

116. Sadeghi, S. and Farhad, A. (2013). An intelligent process planning system based on formal manufacturing capability models. In *Proceedings of the ASME 2013 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE 2013*, pages 1–10.

117. Sang, Z. and Xu, X. (2013). Development of a smart computer numerical control system. In *ASME 2013 International Mechanical Engineering Congress and Exposition*, pages 1–8. American Society of Mechanical Engineers.

118. Santochi, M. and Dini, G. (1996). Use of neural networks in automated selection of technological parameters of cutting tools. *Computer Integrated Manufacturing Systems*, 9(3):137–148.

119. Schenck, D. A. and Wilson, P. R. (1994). *Information modeling: the EXPRESS way*. Oxford University Press, Inc.

120. Seiger, R., Seidl, C., Aßmann, U., and Schlegel, T. (2015). A capability-based framework for programming small domestic service robots. In *Proceedings of the 2015 Joint MORSE/VAO Workshop on Model-Driven Robot Software Engineering and View-based Software-Engineering*, pages 49–54. ACM.

121. SET (1989). Industrial automation - external representation of product definition data - data exchange and transfer standard specification. *L'Association Francaise de Normalisation*.

122. Shen, W., Wang, L., and Hao, Q. (2006). Agent-based distributed manufacturing process planning and scheduling: a state-of-the-art survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 36(4):563–577.

123. Shi, Y. and Eberhart, R. C. (1999). Empirical study of particle swarm optimization. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 3. IEEE.

124. Shunmugam, M., Reddy, S. B., and Narendran, T. (2000). Selection of optimal conditions in multi-pass face-milling using a genetic algorithm. *International Journal of Machine Tools and Manufacture*, 40(3):401–414.

125. Siemens NX (2015). Accessed 26 feb 2017. *http://www.plm.automation.siemens.com*.

126. Smith, B., Brauner, K., Kennicott, P., Liewald, M., and Wellington, J. (1983). Initial graphics exchange specification(iges) version 2. 0. *NTIS, SPRINGFIELD, VA(USA), 1983, 341*.

127. Srinivasan, M. and Sheng, P. (1999). Feature based process planning in environmentally conscious machining–part 2: macroplanning. *Robotics and Computer-Integrated Manufacturing*, 15(3):271–281.

128. Stephenson, D. A. and Agapiou, J. S. (2016). *Metal cutting theory and practice*. CRC press.

129. STEPMAN Project (2015). *The European Commission*. The Seventh Framework Programme.

130. Stewart, D. (1965). A platform with six degrees of freedom. *Proceedings of the institution of mechanical engineers*, 180(1):371–386.

131. Stock, T. and Seliger, G. (2016). Opportunities of sustainable manufacturing in industry 4.0. *Procedia CIRP*, 40:536–541.

132. Su, Y., Chu, X., Chen, D., and Sun, X. (2015). A genetic algorithm for operation sequencing in capp using edge selection based encoding strategy. *Journal of Intelligent Manufacturing*, pages 1–20.

133. Suh, S.-H., Lee, B.-E., Chung, D.-H., and Cheon, S. (2003a). Architecture and implementation of a shop-floor programming system for step-compliant cnc. *Computer-Aided Design*, 35(12):1069–1083.

134. Suh, S.-H., Seo, Y., Lee, S.-M., Choi, T.-H., Jeong, G.-S., and Kim, D.-Y. (2003b). Modelling and implementation of internet-based virtual machine tools. *The International Journal of Advanced Manufacturing Technology*, 21(7):516–522.

135. Tan, W. and Khoshnevis, B. (2000). Integration of process planning and schedulingâĂŤa review. *Journal of Intelligent Manufacturing*, 11(1):51–63.

136. Tanaka, F., Onosato, M., Kishinami, T., Akama, K., Yamada, M., Kondo, T., and Mistui, S. (2008). Modeling and implementation of digital semantic machining models for 5-axis machining application. In *Manufacturing Systems and Technologies for the New Frontier*, pages 177–182. Springer.

137. Vandenbrande, J. H. and Requicha, A. A. (1993). Spatial reasoning for the automatic recognition of machinable features in solid models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(12):1269–1285.

138. VDI/DGQ 3441 (1977). Statistical examination of exact working and positioning of machine tools.

139. Vericut CGTech (2017). Accessed 26 feb 2017. *http://www.cgtech.com/ products/about-vericut/machine-simulation/.*

140. Versprille, K. (2005). Dassault systemes' strategic initiative: 3d xml for sharing product information. *Technology Trends in PLM, Collaborative Product Development Associates, http://www. 3ds. com/uploads/tx_ user3dsplmxml/3DXML_for_ sharing_-product_ information. pdf.*

141. Vichare, P., Nassehi, A., and Newman, S. (2009). A unified manufacturing resource model for representation of computerized numerically controlled machine tools. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 223(5):463–483.

142. Vichare, P., Nassehi, A., Thompson, J., Newman, S. T., Wood, F., and Kumar, S. (2015). Machine tool capability profiles for representing machine tool health. *Robotics and Computer-Integrated Manufacturing*, 34:70–78.

143. Wan, M., Liu, Y., and Zhang, W. (2016). A new algorithm for the identification of cnc geometric errors. *Procedia CIRP*, 56:293–298.

144. Wang, L. (2013). Machine availability monitoring and machining process planning towards cloud manufacturing. *CIRP Journal of Manufacturing Science and Technology*, 6(4):263–273.

145. Wang, S., Wan, J., Zhang, D., Li, D., and Zhang, C. (2016). Towards smart factory for industry 4.0: A self-organized multi-agent system with big data based feedback and coordination. *Computer Networks*, 101:158–168.

146. Wang, Y., Kung, L., and Byrd, T. A. (2018). Big data analytics: Understanding its capabilities and potential benefits for healthcare organizations. *Technological Forecasting and Social Change*, 126:3–13.

147. Weck, M. and Staimer, D. (2002a). Accuracy issues of parallel kinematic machine tools. *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics*, 216(1):51–57.

148. Weck, M. and Staimer, D. (2002b). Parallel kinematic machine tools–current state and future potentials. *CIRP Annals-Manufacturing Technology*, 51(2):671–683.

149. Wesley, M. and Markowsky, G. (1981). Fleshing out projections. *IBM Journal of Research and Development*, 25(6):934–954.

150. Whittaker, E. T. (1988). *A treatise on the analytical dynamics of particles and rigid bodies*. Cambridge University Press.

151. Wilczynski, D. and Lipkis, T. (1993). Machine modelling in loom. *LOOM papers*.

152. Wysk, R. A. (1977). *An automated process planning and selection program: APPAS*. PhD thesis, Purdue University.

153. Xu, X. (2006). Realization of step-nc enabled machining. *Robotics and Computer-Integrated Manufacturing*, 22(2):144–153.

154. Xu, X., Wang, L., and Newman, S. T. (2011). Computer-aided process planning–A critical review of recent developments and future trends. *International Journal of Computer Integrated Manufacturing*, 24(1):1–31.

155. Yang, J., Yuan, J., and Ni, J. (1999). Thermal error mode analysis and robust modeling for error compensation on a cnc turning center. *International Journal of Machine Tools and Manufacture*, 39(9):1367–1381.

156. Yue, Y., Ding, L., Ahmet, K., Painter, J., and Walters, M. (2002). Study of neural network techniques for computer integrated manufacturing. *Engineering Computations*, 19(2):136–157.

157. Yuen, A., Zhang, K., and Altintas, Y. (2013). Smooth trajectory generation for five-axis machine tools. *International Journal of Machine Tools and Manufacture*, 71:11–19.

158. Yun, W. S., Kim, S. K., and Cho, D. W. (1999). Thermal error analysis for a cnc lathe feed drive system. *International Journal of Machine tools and manufacture*, 39(7):1087–1101.

159. Yusof, Y. and Case, K. (2010). Design of a step compliant system for turning operations. *Robotics and Computer-Integrated Manufacturing*, 26(6):753–758.

160. Yusof, Y. and Latif, K. (2014). Survey on computer-aided process planning. *The International Journal of Advanced Manufacturing Technology*, 75(1-4):77–89.

161. Yusof, Y., Newman, S. T., Nassehi, A., and Case, K. (2009). Interoperable cnc system for turning operations. *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, 3(1):112–118.

162. Zahid, M. O., Case, K., and Watts, D. (2014). Cutting tools in finishing operations for cnc rapid manufacturing processes: Experimental studies. *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, 8(6):1123–1127.

163. Zhang, F. (1997). Genetic algorithm in computer-aided process planning. *MEng thesis, National University of Singapore.*

164. Zhang, X., Nassehi, A., and Newman, S. T. (2014). Feature recognition from CNC part programs for milling operations. *The International Journal of Advanced Manufacturing Technology*, pages 1–16.

165. Zhang, Y., Feng, S. C., Wang, X., Tian, W., and Wu, R. (1999). Object oriented manufacturing resource modelling for adaptive process planning. *International Journal of Production Research*, 37(18):4179–4195.

166. Zhang, Y., Fu, J.-z., and Chen, Z.-c. (2013). Machining tests to identify kinematic errors of machine tool table rotation axis based on sensitive directions. *The International Journal of Advanced Manufacturing Technology*, 67(1-4):495–500.

167. Zhao, F., Xu, X., and Xie, S. (2008). Step-nc enabled on-line inspection in support of closed-loop machining. *Robotics and Computer-Integrated Manufacturing*, 24(2):200–216.

Appendix A

A sample STEP-NC file to finish a rectangular pocket

```
#1=PROJECT ('EXCECUTE EXAMPLE1', #2, (#3), $, $, $);
#2=WORKPLAN ('MAIN WORKPLAN', (#10, #11, #12, #13, #14), $, #6, $);
#3=WORKPIECE ('SIMPLE WORKPIECE', #4, 0.010, $, $, $, (#66, #67,
#68, #69));
#4=MATERIAL ('ST-50', 'STEEL', (#5));
#5=PROPERTY_PARAMETER ('E=200000N/M2');
#6=SETUP ('SETUP1', #71, #62, (#7));
#7=WORKPIECE_SETUP (#3, #74, $, $, ());
.
.
#13=MACHINING_WORKINGSTEP ('WS ROUGH POCKET1', #62, #18, #22, $);
.
.
#18=CLOSED_POCKET ('POCKET1', #4, (#22, #23), #84, #65, (), $, #27
, #35, #37, #28);
.
.
#22=BOTTOM_AND_SIDE_ROUGH_MILLING ($, $, 'ROUGH POCKET1', 15.000,
$, #39, #50, #41, $, $, $, #51, 2.500, 5.000, 1.000, 0.500);
.
.
#35=TOLERANCED_LENGTH_MEASURE (80.000, #36);
#36=PLUS_MINUS_VALUE (0.100, 0.100, 3);
#37=TOLERANCED_LENGTH_MEASURE (50.000, #38);
#38=PLUS_MINUS_VALUE (0.100, 0.100, 3);
.
.
#39=MILLING_CUTTING_TOOL ('MILL 20MM', #29, (#125), 80.000, $, $);
```

Appendix B

A sample MCP file for Bridgeport milling machine

Figure B-1: Bridgeport milling machine

Table B.1: Bridgeport milling machine specifications

Machine	Envelope (mm)	Feed (ipm)	Speed (rpm)
VMC 610 XP2	$609 \times 508 \times 609$	1700	24000

```
ISO-10303-21
HEADER;
FILE_DESCRIPTION(
('STEPMAN generated STEP file'),
'2;1');
FILE_NAME(
'TEST.STP',
('Behnood Afshari','Wesley Essink','Aydin Nassehi'),
('University of Bath'),
' ',
'STEPMAN',
' ');
FILE_SCHEMA(('COMBINED_Schema'));
ENDSEC;
DATA;
#1=MACHINING_CAPABILITY_PROFILE(#23,(#2));
#2=MACHINING_CAPABILITY_DATA_POINT(
'"Bridgeport VMC 610 XP2 capability
test"',(#5,#6,#7,#8,#9,#10),#3,#4);
#3=CALENDAR_DATE(2013,31,7);
#4=LOCAL_TIME(16,0,0.0,$);
#5=FEED_DRIVE_CAPABILITY($,0.1,0.0,0.1,$,$);
#6=SPINDLE_DRIVE_CAPABILITY($,$,0.1,10.0,8000.0);
#7=TOOLING_CAPABILITY(13.0,66.0,30.0,(#11,#14,#17,#20));
#8=WORKING_AREA_CAPABILITY(210.0,100.0,200.0,$,$,$,$,$,$,500.0);
#9=OVERALL_CAPABILITY(.MILLING_MACHINE.,#35,3.0);
#10=AXES_CAPABILITY($,(#36));
#11=CUTTING_TOOL_DICTIONARY($,$,$,(),'"rough mill"',#12,150.0);
#12=MILLING_TOOL_BODY(#13,2,$,$,$);
#13=MILLING_TOOL_DIMENSION(40.0,$,$,$,$,$,$);
#14=CUTTING_TOOL_DICTIONARY($,$,$,(),'"end mill"',#15,170.0);
#15=MILLING_TOOL_BODY(#16,2,$,$,$);
#16=MILLING_TOOL_DIMENSION(40.0,$,$,$,$,$,$);
#17=CUTTING_TOOL_DICTIONARY($,$,$,(),'"end mill"',#18,180.0);
#18=MILLING_TOOL_BODY(#19,2,$,$,$);
#19=MILLING_TOOL_DIMENSION(10.0,$,$,$,$,$,$);
#20=CUTTING_TOOL_DICTIONARY($,$,$,(),'"Drill bit"',#21,180.0);
#21=MILLING_TOOL_BODY(#22,2,$,$,$);
#22=MILLING_TOOL_DIMENSION(20.0,$,$,$,$,$,$);
#23=MACHINE_TOOL_SPECIFICATION('"Bridgeport VMC 610
XP2"',.MILLING_MACHINE.,#27,(#28),$,$,#24,
#29,$,(#30,#31,#32,#33));
#24=INSTALLATION(500.0,#25,#26,$,$,$);
#25=MACHINE_SIZE(3025.0,2160.0,2675.0);
#26=ELECTRICAL(3,12500.0,50.0,'"N/A"','"N/A"',220.0);
#27=DEVICE_ID('"Bridgeport"','"VMC 610 XP2"','"723703"',
'"Hardinge Group"',$);
#28=MACHINING_CAPABILITY();
#29=NC_CONTROLLER('"SINUMERIK"','"Siemens"',
.INCH_AND_METRIC.,5.0,4.0,
1.0,0.001,0.01,2.0,(),(),$,$,$,$,(),(),(),$,$);
#30=MACHINE_TOOL_ELEMENT('"X AXIS"',$,$,());
#31=MACHINE_TOOL_ELEMENT('"Y AXIS"',$,$,());
#32=MACHINE_TOOL_ELEMENT('"Z AXIS"',$,$,());
#33=MACHINE_TOOL_ELEMENT('"Work table"',$,$,(#34));
#34=RECTANGULAR_WORK_TABLE('"Rectangular work table"',
```

```
.F.,$,$,$,$,470.0,840.0);
#35=NC_SCHEMA($,$,$,$);
#36=AXES_ACCURACY($,$,#37,#38,#39,$,$,$);
#37=MOVEMENT_ALONG_X($,$,0.05,$);
#38=MOVEMENT_ALONG_Y($,$,0.04,$);
#39=MOVEMENT_ALONG_Z($,$,0.05,$);
ENDSEC;
END-ISO-10303-21
```

Appendix C

A sample STEP-NC code to drill a hole

```
#1=PROJECT ('EXECUTE EXAMPLE 1', #2 , (#3), $, $, $);
#2=WORKPLAN ('MAIN WORKPLAN', (#10, #11, #12, #13, #14), $, #6, $);
#3=WORKPIECE ('SIMPLE WORKPIECE', #4, 0.010, $, $, $,
(#66, #67, #68, #69));
#4=MATERIAL ('ST-50', 'STEEL', (#5));
#5=PROPERTY_PARAMETER ('E=200000N/M2');
#6=SETUP ('SETUP 1', #71, #62, (#7));
#7=WORKPIECE_SETUP (#3, #74, $, $, ());
.
.
#11=MACHINING_WORKINGSTEP ('WS DRILL HOLE 1', #62, #17, #20, $);
.
.
#17=ROUND_HOLE ('HOLE 1 D=22M' , #4, (#20, #21), #81, #64, #56,
$, #26);
.
.
#20=DRILLING ($, $, 'DRILL HOLE 1', 10.000, $, #44, #45, #41, $ ,$,
$,$,$, #46);
.
.
#56=TOLERNACED_LENGTH_MEASURE (22.000, #57);
#57=PLUS_MINUS_VALUE (0.0100, 0.0100, 3);
.
.
#44=MILLING_CUTTING_TOOL ('SPIRAL DRILL 20MM', #31, (#126),
90.000, $, $);
.
.
```

```
#66=CARTESIAN_POINT ('CLAMPING POSITION 1',
(0.000, 40.000,50.000));
#67=CARTESIAN_POINT ('CLAMPING POSITION 2',
(100.000, 40.000,50.000));
#68=CARTESIAN_POINT ('CLAMPING POSITION 3',
(0.000, 200.000,50.000));
#69=CARTESIAN_POINT ('CLAMPING POSITION 4',
(100.000, 200.000, 50.000));
```

Appendix D

The MCP file for XYZ milling machine

Figure D-1: XYZ milling machine

Table D.1: XYZ milling machine specifications

Machine	Envelope (mm)	Feed (mm/min)	Speed (rpm)
XYZ 1020 VMC	1020 × 520 × 546	10000	8000

```
ISO-10303-21
HEADER;
FILE_DESCRIPTION(
('STEPMAN generated STEP file'),
'2;1');
FILE_NAME(
'TEST.STP',
('Behnood Afshari','Wesley Essink','Aydin Nassehi'),
('University of Bath'),
' ',
'STEPMAN',
' ');
FILE_SCHEMA(('COMBINED_Schema'));
ENDSEC;
DATA;
#1=MACHINING_CAPABILITY_PROFILE(#20,(#2));
#2=MACHINING_CAPABILITY_DATA_POINT(
'"XYZ milling machine capability
test"',(#5,#6,#7,#8,#9,#10),#3,#4);
#3=CALENDAR_DATE(2013,31,7);
#4=LOCAL_TIME(16,0,0.0,$);
#5=FEED_DRIVE_CAPABILITY($,20.0,0.0,0.005,$,$);
#6=SPINDLE_DRIVE_CAPABILITY($,$,0.005,60.0,8000.0);
#7=TOOLING_CAPABILITY(300.0,80.0,24.0,(#11,#14,#17));
#8=WORKING_AREA_CAPABILITY(1020.0,520.0,
546.0,$,$,$,$,$,$,500.0);
#9=OVERALL_CAPABILITY(.MILLING_MACHINE.,#32,3.0);
#10=AXES_CAPABILITY($,(#33));
#11=CUTTING_TOOL_DICTIONARY($,$,$,(),'"rough mill"',#12,150.0);
#12=MILLING_TOOL_BODY(#13,2,$,$,$);
#13=MILLING_TOOL_DIMENSION(66.0,$,$,$,$,$,$);
#14=CUTTING_TOOL_DICTIONARY($,$,$,(),'"end mill"',#15,170.0);
#15=MILLING_TOOL_BODY(#16,2,$,$,$);
#16=MILLING_TOOL_DIMENSION(66.0,$,$,$,$,$,$);
#17=CUTTING_TOOL_DICTIONARY($,$,$,(),'"end mill"',#18,180.0);
#18=MILLING_TOOL_BODY(#19,2,$,$,$);
#19=MILLING_TOOL_DIMENSION(20.0,$,$,$,$,$,$);
#20=MACHINE_TOOL_SPECIFICATION('"XYZ 3-axis milling
machine"',.MILLING_MACHINE.,#24,(#25),$,$,#21,
#26,$,(#27,#28,#29,#30));
#21=INSTALLATION(5800.0,#22,#23,$,$,$);
#22=MACHINE_SIZE(2650.0,2105.0,3020.0);
#23=ELECTRICAL(3,12500.0,50.0,'"N/A"','"N/A"',220.0);
#24=DEVICE_ID('"XYZ"','"Vertical milling machine"','"723703"',
'"Hardinge Group"',$);
#25=MACHINING_CAPABILITY();
#26=NC_CONTROLLER('"SINUMERIK"','"Siemens"',
.INCH_AND_METRIC.,5.0,4.0,1.0,0.001,
0.01,2.0,(),(),$,$,$,$,(),(),(),$,$);
#27=MACHINE_TOOL_ELEMENT('"X AXIS"',$,$,());
#28=MACHINE_TOOL_ELEMENT('"Y AXIS"',$,$,());
#29=MACHINE_TOOL_ELEMENT('"Z AXIS"',$,$,());
#30=MACHINE_TOOL_ELEMENT('"Work table"',$,$,(#31));
#31=RECTANGULAR_WORK_TABLE('"Rectangular work table"',
.F.,$,$,$,$,$,500.0,1120.0);
#32=NC_SCHEMA($,$,$,$);
```

```
#33=AXES_ACCURACY($,$,#34,#35,#36,$,$,$);
#34=MOVEMENT_ALONG_X($,$,0.05,$);
#35=MOVEMENT_ALONG_Y($,$,0.04,$);
#36=MOVEMENT_ALONG_Z($,$,0.05,$);
ENDSEC;
END-ISO-10303-21
```

Appendix E

The MCP file for Renishaw PKM platform

Figure E-1: Renishaw PKM machine

Table E.1: Renishaw PKM machine specifications

Machine	Envelope (mm)	Feed (mm/min)	Speed (rpm)
Equator 300	$300 \times 300 \times 150$	8000	50000

```
HEADER;
FILE_DESCRIPTION(('STEPMAN generated STEP file'),
'2;1');
FILE_NAME('TEST.STP',('Behnood Afshari','Wesley Essink',
'Aydin Nassehi'), ('University of Bath'),' ','STEPMAN',' ');
FILE_SCHEMA(('COMBINED_Schema'));
ENDSEC;
```

```
DATA;
#1=MACHINING_CAPABILITY_PROFILE(#22,(#2));
#2=MACHINING_CAPABILITY_DATA_POINT('"PKM PLATFORM TEST"',(#5, #6, #7,
#8, #9, #10);
#3=CALENDAR_DATE(2016,31,7);
#4=LOCAL_TIME(16,0,0.0,$);
#5=KINEMATIC_CAPABILITY(#11);
#6=FEED_DRIVE_CAPABILITY($,0.1,0.0,0.1,$);
#7=SPINDLE_DRIVE_CAPABILITY($,$,0.1,10.0,8000.0);
#8=TOOLING_CAPABILITY(13.0,66.0,30.0,(#18,#21,#24));
#9=WORKING_AREA_CAPABILITY(210.0,100.0,200.0,$,$,$,$,$,$,500.0);
#10=OVERALL_CAPABILITY(.MILLING_MACHINE.,#39,3.0);
#11=PARALLEL_KINEMATIC('"RENISHAW EQUATOR PROFILE"',
(#12, #13, #14),500.0,550.0,600.0,150.0,100.0);
#12=PARALLEL_LEG('"LEG 1"', 478.0,0.06693528,60,35,7.071,7.071);
#13= PARALLEL_LEG('"LEG 2"',478.0,0.02918999,60,35,24.571,37.38);
#14= PARALLEL_LEG('"LEG 3"',478.0,-0.016452,60,35,42.071,7.071);
#15=MOVEMENT_ALONG_X(0.05,0.05,0.05,0.05,5.0E-5,0.05,0.05);
#16=MOVEMENT_ALONG_Y(0.05,0.05,0.05,0.05,5.0E-5,0.05,0.05);
#17=MOVEMENT_ALONG_Z(0.05,0.05,0.05,0.05,5.0E-5,0.05,0.05);
#18=CUTTING_TOOL_DICTIONARY($,$,$,(),'"rough mill"',#19,150.0);
#19=MILLING_TOOL_BODY(#20,2,$,$,$);
#20=MILLING_TOOL_DIMENSION(8.0,$,$,$,$,$,$);
#21=CUTTING_TOOL_DICTIONARY($,$,$,(),'"end mill"',#22,170.0);
#22=MILLING_TOOL_BODY(#23,2,$,$,$);
#23=MILLING_TOOL_DIMENSION(8.5,$,$,$,$,$,$);
#24=CUTTING_TOOL_DICTIONARY($,$,$,(),'"end mill"',#25,180.0);
#25=MILLING_TOOL_BODY(#26,2,$,$,$);
#26=MILLING_TOOL_DIMENSION(10.0,$,$,$,$,$,$);
#27=MACHINE_TOOL_SPECIFICATION('"RENISHAW EQUATOR"'
,.MILLING_MACHINE.,#31,(#32),$,$,#28,#33,$,(#34,#35,#36,#37));
#28=INSTALLATION(500.0,#29,#30,$,$,$);
#29=MACHINE_SIZE(550.0,500.0,600.0);
#30=ELECTRICAL(3,12500.0,50.0,'"N/A"','"N/A"',220.0);
#31=DEVICE_ID('"RENISHAW"','"PKM"'
,'"723703"','"RENISHAW PLC"',$);
#32=MACHINING_CAPABILITY();
#33=NC_CONTROLLER('"N/A"','"N/A"',.INCH_AND_METRIC.
,5.0,4.0,1.0,0.0010,0.01,2.0,(),(),$,$,$,$,(),(),(),$,$);
#34=MACHINE_TOOL_ELEMENT('"X AXIS"',$,$,());
#35=MACHINE_TOOL_ELEMENT('"Y AXIS"',$,$,());
#36=MACHINE_TOOL_ELEMENT('"Z AXIS"',$,$,());
#37=MACHINE_TOOL_ELEMENT('"Work table"',$,$,(#38));
#38=RECTANGULAR_WORK_TABLE('"Rectangular work table"'
,.F.,$,$,$,$,470.0,500.0);
#39=NC_SCHEMA($,$,$,$);
ENDSEC;
END-ISO-10303-21
```

Appendix F

STEP-NC code for manufacturing a part on the PKM platform

```
HEADER;
FILE_DESCRIPTION(('TEST PIECE',
'SIMPLE PROGRAM WITH A POCKET'),'1');
FILE_NAME('MCPTEST.STP',
'2016-02-02',
('BEHNOOD AFSHARIZAND'),
('MECHANICAL DEPARTMENT, UNIVERSITY OF BATH'),
$, 'ISO 14649',$);
FILE_SCHEMA(('MACHINING_SCHEMA','MILLING_SCHEMA'));
ENDSEC;
DATA;
#1= PROJECT('EXECUTE EXAMPLE1',#2,(#4),$,$,$);
#2= WORKPLAN('MAIN WORKPLAN',(#10,#11,#12,#13,#14),$,#8,$);
#4= WORKPIECE('SIMPLE WORKPIECE',#6,0.010,$,$,$,(#66,#67,#68,#69));
#6= MATERIAL('EPX','PLASTIC',(#7));
#7= PROPERTY_PARAMETER(' ');
#8= SETUP('SETUP1',#71,#62,(#9));
#9= WORKPIECE_SETUP(#4,#74,$,$,());
#13= MACHINING_WORKINGSTEP('MACHINE A CLOSED POCKET',#62,#18,#22,$);
#18= CLOSED_POCKET('POCKET1',#4,(#22),#84,#65,(),$,#27,#35,#37,#28);
#22= BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'ROUGH
POCKET1',15.000,$,#39,#50,#41 ,$,$,$,#51,2.500,5.000,1.000,0.500);
#27= PLANAR_POCKET_BOTTOM_CONDITION();
#28= GENERAL_CLOSED_PROFILE($,#59);
#29= TAPERED_ENDMILL(#30,4,$,.F.,$,$);
#30= MILLING_TOOL_DIMENSION(3.000,$,$,$,1.500,$,$);
#35= TOLERANCED_LENGTH_MEASURE(1.000,#36);
#36= PLUS_MINUS_VALUE(0.100,0.100,3);
#37= TOLERANCED_LENGTH_MEASURE(10.000,#38);
#38= PLUS_MINUS_VALUE(0.100,0.100,3);
#39= MILLING_CUTTING_TOOL('MILL 3MM',#29,(#125),12.000,$,$);
#41= MILLING_MACHINE_FUNCTIONS(.T.,$,$,.F.,$,(),.T.,$,$,());
#50= MILLING_TECHNOLOGY($,.TCP.,$,20.000,$,.F.,.F.,.F.,$);
#51= CONTOUR_BIDIRECTIONAL($,$,$,$,$,$);
```

```
#59= POLYLINE('CONTOUR OF POCKET1',(#121,#122,#123,#124,#121));
#62= ELEMENTARY_SURFACE('SECURITY PLANE',#73);
#65= ELEMENTARY_SURFACE('DEPTH SURFACE FOR POCKET',#94);
#66= CARTESIAN_POINT('CLAMPING_POSITION1',(0.000,20.000,25.000));
#67= CARTESIAN_POINT('CLAMPING_POSITION2',(100.000,20.000,25.000));
#68= CARTESIAN_POINT('CLAMPING_POSITION3',(0.000,100.000,25.000));
#69= CARTESIAN_POINT('CLAMPING_POSITION4',(100.000,100.000,25.000));
#71= AXIS2_PLACEMENT_3D('SETUP1',#95,#96,#97);
#73= AXIS2_PLACEMENT_3D('PLANE1',#98,#99,#100);
#74= AXIS2_PLACEMENT_3D('WORKPIECE',#101,#102,#103);
#84= AXIS2_PLACEMENT_3D('POCKET1',#115,#116,#117);
#94= AXIS2_PLACEMENT_3D('POCKET1',#118,#119,#120);
#95= CARTESIAN_POINT('SETUP1:  LOCATION ', (150.000,90.000,40.000));
#96= DIRECTION('AXIS', (0.000,0.000,1.000));
#97= DIRECTION(' REF_DIRECTION',(1.000,0.000,0.000));
#98= CARTESIAN_POINT('SECPLANE1:  LOCATION',(0.000,0.000,30.000));
#99= DIRECTION(' AXIS ',(0.000,0.000,1.000));
#100= DIRECTION(' REF_DIRECTION',(1.000,0.000,0.000));
#101= CARTESIAN_POINT('WORKPIECE1:LOCATION ',(0.000,0.000,0.000));
#102= DIRECTION(' AXIS ',(0.000,0.000,1.000));
#103= DIRECTION(' REF_DIRECTION',(1.000,0.000,0.000));
#115= CARTESIAN_POINT('POCKET1:LOCATION ',(20.000,65.000,0.000));
#116= DIRECTION(' AXIS ',(0.000,0.000,1.000));
#117= DIRECTION('REF_DIRECTION',(-1.000,0.000,0.000));
#118= CARTESIAN_POINT('POCKET1:DEPTH ',(0.000,0.000,-15.000));
#119= DIRECTION('AXIS ',(0.000,0.000,1.000));
#120= DIRECTION('REF_DIRECTION',(1.000,0.000,0.000));
#121= CARTESIAN_POINT('P1',(0.000,0.000,0.000));
#122= CARTESIAN_POINT('P2',(0.000,80.000,0.000));
#123= CARTESIAN_POINT('P3',(-50.000,80.000,0.000));
#124= CARTESIAN_POINT('P4',(-50.000,0.000,0.000));
#125= CUTTING_COMPONENT(80.000,$,$,$);
ENDSEC;
END-ISO-10303-21;
```

Appendix G

Java code for the PKM experiment

```
package PsoTest;

import java.awt.geom.Point2D;
import java.io.File;
import java.io.FileNotFoundException;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class Main {

static  ArrayList<Double> xn = new ArrayList<>();
static  ArrayList<Double> yn = new ArrayList<>();
static  ArrayList<Double> zn = new ArrayList<>();

public static void main(String[] args) throws FileNotFoundException
    {
// TODO Auto-generated method stub
loadFile();
//Initialising PSO parameters
int nIt = 100; //Number of iterations
int nPop = 1000; //Population size
double w = 1; //Inertia
double wDamp = 0.99; //Damping coefficient
double c1 = 2;  // First coefficient of acceleration
double c2 = 2; //Second coefficient of acceleration

double[] varMin = new double[]{0,0}; // X and Y Minimums for Table
double[] varMax = new double[]{98.5, 94}; // X and Y Maximums for
    Table
```

```
//Create particles
Point2D[] particlePositions = new Point2D[nPop];
Point2D[] particleBestPosition = new Point2D[nPop];
Point2D[] particleVelocities = new Point2D[nPop];
double[] particleFitness = new double[nPop];
double[] particleBestFitness = new double[nPop];

//Create global best
Point2D gPBest = new Point2D.Double(0, 0);
double gFBest = Double.POSITIVE_INFINITY;


//Initialise Particles
for (int i = 0; i < particlePositions.length; i++) {
particlePositions[i] = new Point2D.Double(Math.random() *
    (varMax[0] - varMin[0]), Math.random() * (varMax[1] -
    varMin[1]));
particleFitness[i] = calcFitness(particlePositions[i].getX(),
    particlePositions[i].getY());

particleBestPosition[i] = new
    Point2D.Double(particlePositions[i].getX(),
    particlePositions[i].getY());
particleBestFitness[i] = particleFitness[i];

//Check whether particle is best so far
if (particleBestFitness[i] < gFBest) {
gPBest = new Point2D.Double(particleBestPosition[i].getX(),
    particleBestPosition[i].getY());
gFBest = particleBestFitness[i];
System.out.println("Fitness: " + Math.round (gFBest * 10000.0) /
    10000.0 + " X: " + Math.round (gPBest.getX()* 10000.0) / 10000.0
    + " Y: " + Math.round (gPBest.getY()* 10000.0) / 10000.0);
}
particleVelocities[i] = new Point2D.Double(0, 0);
}
//Main loop
for (int i = 0; i < nIt; i++) {
//Iterate through particles
for (int j = 0; j < particlePositions.length; j++) {
//Update velocities
particleVelocities[j] = new Point2D.Double(w *
    particleVelocities[j].getX()
+ c1 * Math.random() * (particleBestPosition[j].getX() -
    particlePositions[j].getX())
+ c2 * Math.random() * (gPBest.getX() -
    particlePositions[j].getX()),
w * particleVelocities[j].getY()
+ c1 * Math.random() * (particleBestPosition[j].getY() -
    particlePositions[j].getY())
+ c2 * Math.random() * (gPBest.getY() -
    particlePositions[j].getY()));

//Update positions
```

```
particlePositions[j] = new
    Point2D.Double(particlePositions[j].getX() +
    particleVelocities[j].getX(),
particlePositions[j].getY() + particleVelocities[j].getY());

//Update fitness
particleFitness[j] = calcFitness(particlePositions[j].getX(),
    particlePositions[j].getY());

//Check if current position is particles best
if (particleFitness[j] < particleBestFitness[j]) {
particleBestPosition[j] = new
    Point2D.Double(particlePositions[j].getX(),
    particlePositions[j].getY());
particleBestFitness[j] = particleFitness[j];

//Check if current position is global best
if (particleBestFitness[j] < gFBest) {
gPBest = new Point2D.Double(particleBestPosition[j].getX(),
    particleBestPosition[j].getY());
gFBest = particleBestFitness[j];
System.out.print("Iteration " + i + " Fitness: " + Math.round
    (gFBest* 100000.0) / 100000.0 + " X: " + Math.round
    (gPBest.getX()* 100000.0) / 100000.0);
System.out.printf(" Y: %.2f\n", Math.round (gPBest.getY()*
    100000.0) / 100000.0);
}
}

}
w *= wDamp;
}
}

private static double calcFitness(final double x, final double y)
    throws FileNotFoundException {
double fitness = 0;
ArrayList<Double> actualX = Xa(x, y);
ArrayList<Double> actualY = Ya(x, y);
for (int i = 0; i < xn.size(); i++) {
fitness += Math.sqrt(Math.pow((x + xn.get(i)) - actualX.get(i), 2)
    + Math.pow((y + yn.get(i)) - actualY.get(i), 2));
}
return fitness;
}

private static void loadFile() throws FileNotFoundException {
String content = new Scanner(
new
File("/Users/ba278/Desktop/gcode2.rtf")).useDelimiter("\\Z").next();
List<String> result=Parse(content); for (int
    i=0;i<result.size();i++) {
if (result.get(i).contains("X"))
    xn.add(Double.parseDouble(result.get(i+1)));
if (result.get(i).contains("Y"))
    yn.add(Double.parseDouble(result.get(i+1)));
```

```
if (result.get(i).contains("Z"))
   zn.add(Double.parseDouble(result.get(i+1)));
}
}

private static ArrayList<Double> Xa(double x, double y) throws
   FileNotFoundException {
ArrayList<Double> l1=new ArrayList<Double>();
ArrayList<Double> l2=new ArrayList<Double>();
ArrayList<Double> xe=new ArrayList<Double>();
double dl1=-0.02;
double dl2=-0.004;

for (int j=0;j<xn.size();j++) {
l1.add(Math.sqrt(Math.pow(xn.get(j)-70.71,
   2)+Math.pow(yn.get(j)-70.71, 2)+Math.pow(zn.get(j)-600, 2)));
l2.add(Math.sqrt(Math.pow(xn.get(j)-372.21,
   2)+Math.pow(yn.get(j)-70.71, 2)+Math.pow(zn.get(j)-600, 2)));
}
for (int j=0;j<xn.size();j++) {
xe.add(xn.get(j)+(2*dl1*l1.get(j))/603-(2*dl2*l2.get(j))/603);
}
return xe;
}

private static ArrayList<Double> Ya(double x, double y) throws
   FileNotFoundException {
ArrayList<Double> l1=new ArrayList<Double>();
ArrayList<Double> l2=new ArrayList<Double>();
ArrayList<Double> l3=new ArrayList<Double>();

ArrayList<Double> ye=new ArrayList<Double>();

double dl1=-0.02;
double dl2=-0.004;
double dl3=-0.001;

for (int j=0;j<yn.size();j++) {
l1.add(Math.sqrt(Math.pow(xn.get(j)-70.71,
   2)+Math.pow(yn.get(j)-70.71, 2)+Math.pow(zn.get(j)-600, 2)));
l2.add(Math.sqrt(Math.pow(xn.get(j)-372.21,
   2)+Math.pow(yn.get(j)-70.71, 2)+Math.pow(zn.get(j)-600, 2)));
l3.add(Math.sqrt(Math.pow(xn.get(j)-221.46,
   2)+Math.pow(yn.get(j)-331.82, 2)+Math.pow(zn.get(j)-600, 2)));
}
for (int j=0;j<yn.size();j++) {
ye.add(yn.get(j)+(50*dl1*l1.get(j))/26111+
(50*dl2*l2.get(j))/26111-(50*dl3*l3.get(j))/26111);
}
return ye;
}

private static List<String> Parse(String str) {
List<String> output = new ArrayList<String>();
Matcher match =
   Pattern.compile("-?[0.0-900.0]+|[a-z]+|[A-Z]").matcher(str);
```

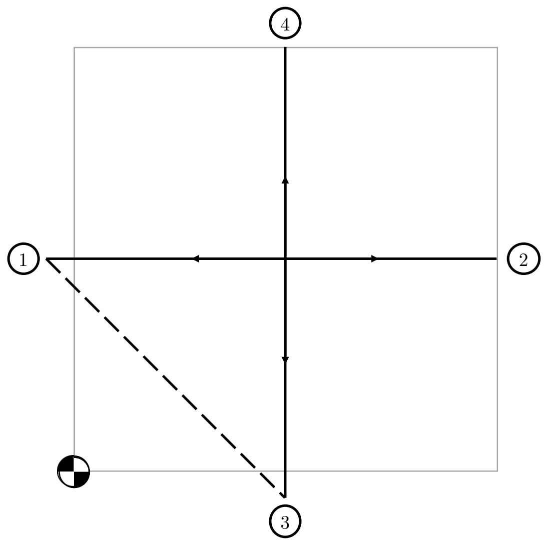
```
    while (match.find()) {
output.add(match.group());
}
return output;
}

}
```

Appendix H

Tool path used for validating CAPPable on Dugard machine tool

```
0002;
G1902 B200 D200 H30 I0 J0 K0;
G21 G80 G40 G90 G17;
T2 D2 M6;
G00 G90 G57 X-10 Y100;
S4000 M3;
G43 Z10 H2;
M8;
G1 Z-5 F500;
X210 F800;
Z-10 F500;
X-10 F800;
G0 Z20;
X100 Y-10;
G1 Z-5 F500;
Y210 F800;
Z-10 F500;
Y-10 F800;
G0 Z100;
M2;
```



Appendix I

The MCP file for Dugard Eagle 850 vertical machining centre

Figure I-1: Dugard Eagle 850 milling machine

Table I.1: Dugard Eagle 850 milling machine specifications

Machine	Envelope (mm)	Feed (mm/min)	Speed (rpm)
Dugard Eagle 850 VMC	850 × 530 × 510	10000	10000

```
 ISO-10303-21 HEADER;
FILE_DESCRIPTION( ('STEPMAN generated STEP file'), '2;
1');
FILE_NAME( 'TEST.STP', ('Behnood Afshari',
'Wesley Essink','Aydin Nassehi'), ('University of Bath'),
' ',
'STEPMAN',
' ');
FILE_SCHEMA(('COMBINED_Schema'));
ENDSEC;
DATA;
#1=MACHINING_CAPABILITY_PROFILE(#44,(#2));
#2=MACHINING_CAPABILITY_DATA_POINT('"Dugard Eagle 850
VMC"',(#5,#6,#7,#8,#9,#10),#3,#4);
#3=CALENDAR_DATE(2018,10,30);
#4=LOCAL_TIME(16,0,0.0,$);
#5=FEED_DRIVE_CAPABILITY($,30.0,0.0,0.1,$,$);
#6=SPINDLE_DRIVE_CAPABILITY($,$,0.1,80.0,10000.0);
#7=TOOLING_CAPABILITY(13.0,76.0,24.0,(#35,#38,#41));
#8=WORKING_AREA_CAPABILITY(850.0,530.0,510.0,$,$,$,$,$,$,500.0);
#9=OVERALL_CAPABILITY(.MILLING_MACHINE.,#56,3.0);
#10=AXES_CAPABILITY(.BALL_BAR.,(#11,#12,#13,#14,
#15,#16,#17,#18));
#11=AXES_ACCURACY(97.1,-97.1,#19,#20,$,125.6,$,$);
#12=AXES_ACCURACY(297.1,-97.1,#21,#22,$,135.4,$,$);
#13=AXES_ACCURACY(497.1,-97.1,#23,#24,$,124.1,$,$);
#14=AXES_ACCURACY(697.1,-97.1,#25,#26,$,137.8,$,$);
#15=AXES_ACCURACY(697.1,-297.1,#27,#28,$,142.1,$,$);
#16=AXES_ACCURACY(497.1,-297.1,#29,#30,$,129.8,$,$);
#17=AXES_ACCURACY(297.1,-297.1,#31,#32,$,142.2,$,$);
#18=AXES_ACCURACY(97.1,-297.1,#33,#34,$,131.8,$,$);
#19=MOVEMENT_ALONG_X($,$,0.0013,$);
#20=MOVEMENT_ALONG_Y($,$,-4.0E-4,$);
#21=MOVEMENT_ALONG_X($,$,1.0E-4,$);
#22=MOVEMENT_ALONG_Y($,$,0.0016,$);
#23=MOVEMENT_ALONG_X($,$,1.0E-4,$);
#24=MOVEMENT_ALONG_Y($,$,0.0017,$);
#25=MOVEMENT_ALONG_X($,$,0.0013,$);
#26=MOVEMENT_ALONG_Y($,$,9.0E-4,$);
#27=MOVEMENT_ALONG_X($,$,6.0E-4,$);
#28=MOVEMENT_ALONG_Y($,$,0.002,$);
#29=MOVEMENT_ALONG_X($,$,-8.0E-4,$);
#30=MOVEMENT_ALONG_Y($,$,0.0025,$);
#31=MOVEMENT_ALONG_X($,$,8.0E-4,$);
#32=MOVEMENT_ALONG_Y($,$,0.0018,$);
#33=MOVEMENT_ALONG_X($,$,4.0E-4,$);
#34=MOVEMENT_ALONG_Y($,$,0.0015,$);
#35=CUTTING_TOOL_DICTIONARY($,$,$,(),'"rough mill"',#36,150.0);
#36=MILLING_TOOL_BODY(#37,2,$,$,$);
#37=MILLING_TOOL_DIMENSION(40.0,$,$,$,$,$,$);
#38=CUTTING_TOOL_DICTIONARY($,$,$,(),'"flat end mill"',#39,170.0);
#39=MILLING_TOOL_BODY(#40,2,$,$,$);
#40=MILLING_TOOL_DIMENSION(16.0,$,$,$,$,$,$);
#41=CUTTING_TOOL_DICTIONARY($,$,$,(),'"end mill"',#42,180.0);
#42=MILLING_TOOL_BODY(#43,2,$,$,$);
#43=MILLING_TOOL_DIMENSION(10.0,$,$,$,$,$,$);
```

```
#44=MACHINE_TOOL_SPECIFICATION('"Dugard Eagle 850
VMC"',.MILLING_MACHINE.,#48,(#49),$,$,#45,#50,$, (#51,#52,#53,#54));
#45=INSTALLATION(5800.0,#46,#47,$,$,$);
#46=MACHINE_SIZE(2300.0,2060.0,2583.0);
#47=ELECTRICAL(3,12500.0,50.0,'"N/A"','"N/A"',220.0);
#48=DEVICE_ID('"Dugard"','"VMC 850"','"723703"','" Dugard"',$);
#49=MACHINING_CAPABILITY();
#50=NC_CONTROLLER('"Fanuc"','"Series
18i"',.INCH_AND_METRIC.,4.0,4.0,1.0,0.001,0.01,
2.0,(),(),$,$,$,$,(),(),(),$,$);
#51=MACHINE_TOOL_ELEMENT('"X AXIS"',$,$,());
#52=MACHINE_TOOL_ELEMENT('"Y AXIS"',$,$,());
#53=MACHINE_TOOL_ELEMENT('"Z AXIS"',$,$,());
#54=MACHINE_TOOL_ELEMENT('"Work table"',$,$,(#55));
#55=RECTANGULAR_WORK_TABLE('"Rectangular work
table"',.F.,$,$,$,$,510.0,1000.0);
#56=NC_SCHEMA($,$,$,$);
ENDSEC;
END-ISO-10303-21
```

Appendix J

List of presented papers from this research

1. Behnood Afsharizand, Aydin Nassehi, Vimal Dhokia and Stephen T. Newman (2014) Formal modelling of process planning in combined additive and subtractive manufacturing, Enabling Manufacturing Competitiveness and Economic Sustainability, Springer International Publishing, pp. 171-176.

2. Behnood Afsharizand, Xhianzhi Zhang, Stephen T. Newman and Aydin Nassehi (2014) Determination of machinability considering degradation of accuracy over machine tool life cycle, Variety Management in Manufacturing, Proceedings of the 47th CIRP Conference on Manufacturing Systems, pp. 760-765.

3. Xhianzhi Zhang, Behnood Afsharizand, Wesley Essink, Stephen T. Newman and Aydin Nassehi (2014) A STEP-compliant method for manufacturing knowledge capture, Procedia CIRP, 20, pp. 103-108.